Elizabeth City State University
ONR-AASERT
Summer 1996 Research Teams

Dr. Linda Bailey Hayden, Principal Investigator

Fractals/Chaos with Mathematica Team
Dr. Manglik, Mentor
Timothy McCray, Graduate Student-CS
Lakesha Mundon, Sophomore-Math
Tammara Ward, Junior- Math
Tanisha Cowell, Junior-CS

ATM Networking Team
Dr. Linda Hayden, Mentor
Mr. Darnley Archer, Mentor
Mr. Wayman White, Mentor
Sharon Saunders, Graduate Student-CS
Derrek Burrus, Sophomore-CS
Shanita Powell, Sophomore-CS
Curtis Felton, Junior - CS/Chem
Antonio Rook, Sophomore-CS

HTML/JAVA
Dr. Linda Hayden, Mentor
Mrs. Tracy Chamberlain, Mentor
Michelle Brown-Emmanual, Graduate Student-CS
Marie Dail, Graduate Student-CS
Kimberly Wright, Sophomore-CS
Kuchumbi Hayden, Sophomore-CS
Reginald Turner, Senior-CS
Courtney Fields, Sophomore-CS
Makeba Fussell, Senior-CS

REPORT DOCUMENTATION PAGE

FORM APPROVED
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Quarterly Report

4. TITLE AND SUBTITLE OF REPORT
ONR Summer 1996 Research Teams

5. FUNDING NUMBERS
N00014-94-1-1089 PP and CV-1
N00014-94-1-0948 N-AASERT-2

6. AUTHOR(S)
Dr. Linda Hayden

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Elizabeth City State University
1704 Weeksville Road
Campus Box 672
Elizabeth City NC 27909

8. PERFORMING ORGANIZATION REPORT NUMBER:
5-52562
5-52561

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research

10. SPONSORING/MONITORING AGENCY REPORT NUMBER:

11. SUPPLEMENTARY NOTES:

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE
N00179

13. ABSTRACT (Maximum 200 words)

The AASERT Summer Research Program is part of a trio of programs at ECSU funded by ONR. They include the parent grant Nurturing ECSU Research Talent (NERT), NERT-I(Instrumentation) and Augmentation Award for Science and Engineering Research Training(AASERT). The AASERT grant provides funds for the summer component while NERT-I provides instrumentation for both NERT and AASERT.

Student development activities have included the following a)Recruitment of high ability minority students;b) Providing a summer program for recruited students;c) Providing research experiences;d) Providing a mentor, graduate school counseling and GRE preparation;e) Providing financial support for students in the form of research assistantships; and f) Providing funds for student travel.

This report documents the summer research activities of the NERT and AASERT program.

14. SUBJECT TERMS	15. NUMBER OF PAGES:
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT:	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

ONRCHI COMPUTER V 1.0 11/94 STANDARD FORM 298 (REV 2-89)

DTIC QUALITY INSPECTED 1

About the Program.....

This program, entitled Nurturing ECSU Research Talent (NERT) focuses on undergraduate education and undergraduate research experiences.. Nurturing these young researchers is our primary concern. Highest priority is given to providing them with the guidance and skills to insure their entrance and success in graduate school. Further, each student in our program learns the fundamentals of scientific research as they conduct investigations in HTML/JAVA, Asynchronous Transfer Mode Networking and Fractals/ Chaos.

AASERT Summer Research program is part of a trio of programs at ECSU funded by ONR. They include the parent grant Nurturing ECSU Research Talent(NERT), NERT-1(Instrumentation) and Augmentation Award for Science and Engineering Research Training(AASERT). The AASERT grant provides funds for the summer component while NERT-1 provides instrumentation for both NERT and AASERT.

Student development activities have included the following: a) Recruitment of high ability minority students b) Providing a summer program for recruited students; c) Providing research experiences; d) Providing a mentor, graduate school counseling and GRE preparation; e) Providing financial support for students in the form of research assistantships; and f) Providing funds for student travel.

This program also strengthens the infrastructure of the Mathematics and Computer Science Department of ECSU. Activities which address infrastructure have included a) Enhancement of current computer graphics and operating systems courses; b) Development of a new courses c) Acquisition of computer equipment appropriate to support of student research; d) Establishing a visiting lecture series in computer science and mathematics; e) Hiring a UNIX network manager.

ECSU is a small school that makes a big effort to nurture their students. I am proud to part of the mentoring effort. It has been my pleasure to work with these young people who are preparing themselves to assume future leadership roles within the technical ranks. May they continue their quest for knowledge and excellence!

Dr. Linda Bailey Hayden,
NERT Principal Investigator

Office of Naval Research
AASERT Summer'96 Research Program
June 24. 1996 - August 2, 1996

Dr. Linda Hayden, Principal Investigator

This ONR-AASERT research project, at ECSU, supports undergraduates and precollege students in our summer research training. All students hired under this research project investigate a mathematics or computer science topic. Each will also develop a personal Homepage.

Undergraduate Computer Science majors must be full time ECSU students with a minimum 2.75 overall GPA, 3.0 GPA in their major courses and must be recommended by two of their major professors. The undergraduates will work in the laboratory for 6 hours each day, 5 days each week for 6 weeks.

Precollege students selected have completed a minimum of three credits of mathematics including geometry and algebra II. Grades of B or better in these courses plus recommendation of two science/mathematics teachers will be required. The precollege students will work in the laboratory for five weeks, 6 hours each day, 5 days each week. All students, both precollege and undergraduate must be citizens of the United States.

Student Salaries: Precollege students receive $7.00/hr. Undergraduate students get $8.00/hr.

<u>Planned Activities</u>

- •Lectures by visiting consultants
- •Bi-weekly Progress Reports: Fridays 1:00pm - 2:30pm
- •Final Research Project Reports
  - Final Oral Reports and Final Written Reports: Aug. 2, 1996
- •Conference Travel
  - ADMI conference Mayaquez, Puerto Rico, July 25-28, 1996
- •Faculty Mentors
- •Graduate School Assistants

Summer 1996 Research Teams
Elizabeth City State University

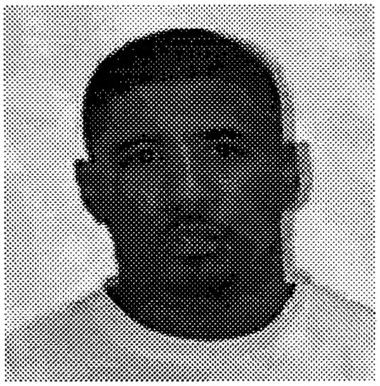
Dr. Linda Hayden, Principal Investigator

TEAM NAME	MENTOR	GRAD STUDENT(S)	ECSU STUDENTS
Fractals/Chaos with Mathematica	Dr. Manglik √	Timothy McCray **	Tammara Ward √ Lakisha Mundon *
HTML/JAVA	Mrs.Tracy Chamberlain	Marie Dail Michelle Brown **	Courtney Fields* Reginald Turner √√ Kimberly Wright*** Makeba Fussel √√ Kuchumbi Hayden *
ATM Networks	Mr. Darnley Archer Mr. Wayman White	Sharon Saunders **	Antonio Rook *. Curtis Felton √ Derrek Burrus √ Vara Powell √

Contract Dates

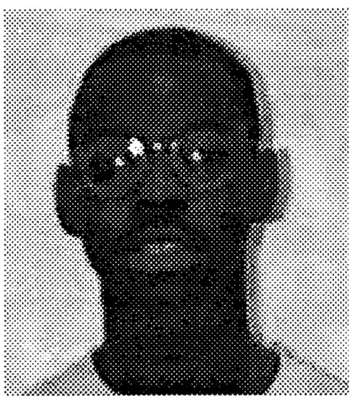
* May 13-Aug 2 **May 20 - Aug 2 *** May 7 - Jul 19 √ June 24 - Aug 2 √√ June 24 - July 19

1996 SUMMER RESEARCHERS

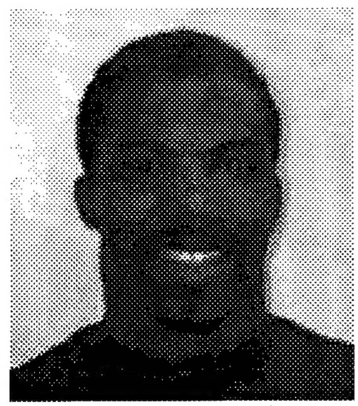


Antonio Rook

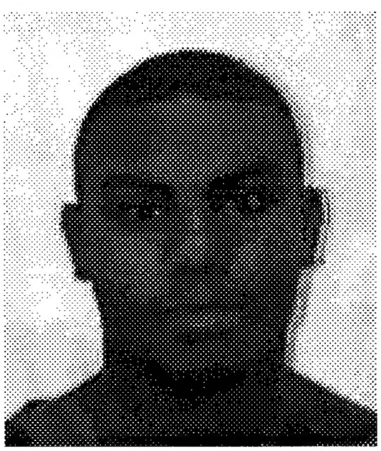
Courtney Fields



Curtis Felton

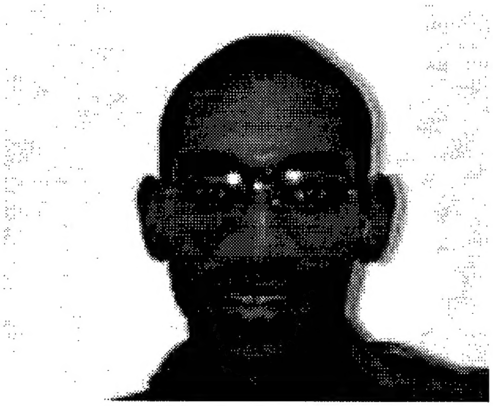


Darnley Archer
Mentor



Derrek Burrus

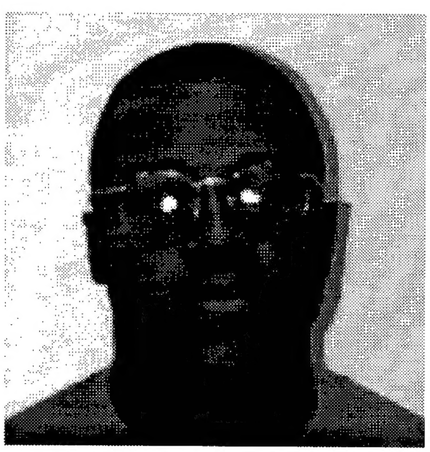
Reginald Turner



Wayman White
Mentor

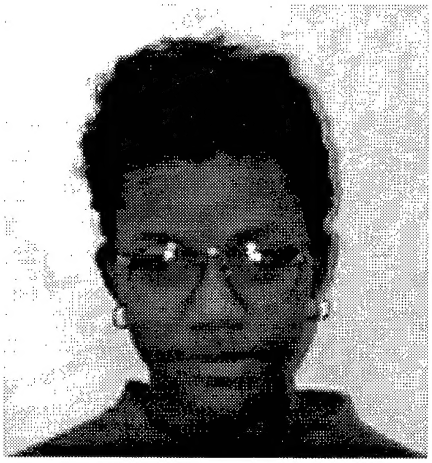
Dr.Vinod Manglik
Mentor

Tracy Chamberlain
Mentor



Timothy McCray
Graduate Student

Sharon Saunders
Graduate Student

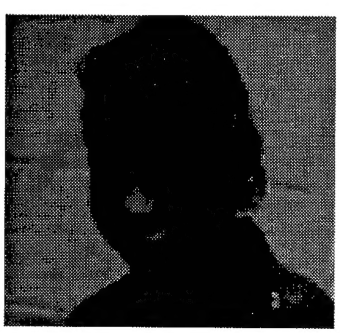


Shanita Powell

Marie Dail
Graduate Student

Kuchumbi Hayden

Tammara Ward



Kimberly Wright

Tanisha Cowell

1996 SUMMER RESEARCH GROUPS

Back row: Courtney Fields, Reginald Turner, Kuchumbi Hayden
Front Row: Tracy Chamberlain, Makeba Fussell, Michelle Brown-Emmanual

1996 SUMMER RESEARCH GROUPS

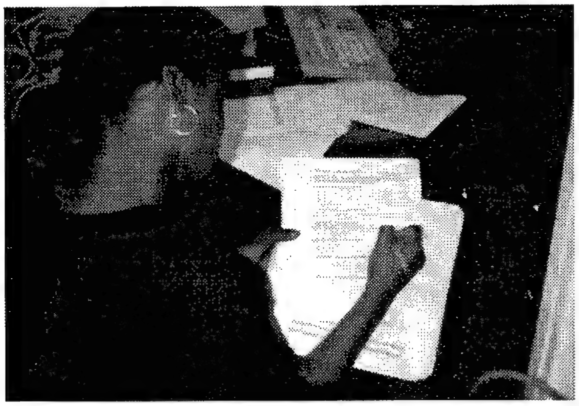
Tanisha Cowell, Timothy McCray, Tammara Ward
No Photo: Lakesha Mundon, Dr. Manglik

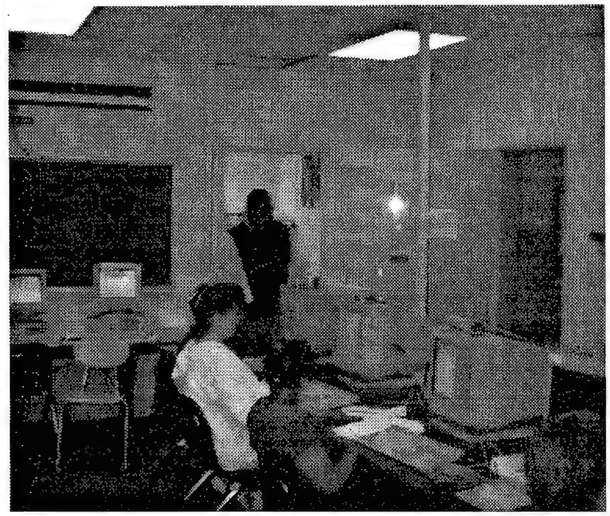
Back Row: Wayman White, Shanita Powell, Curtis Felton, Antonio Rook
Front Row: Derrek Burrus, Sharon Saunders, Darnley Archer

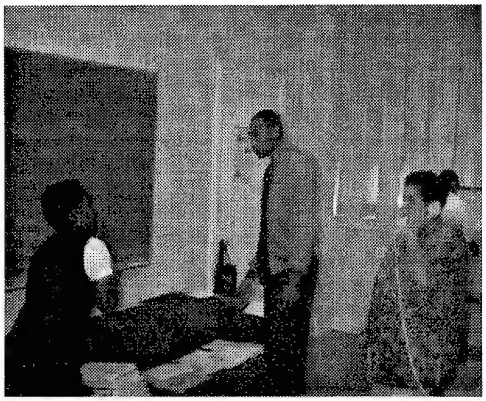
1996 Summer AASERT Program

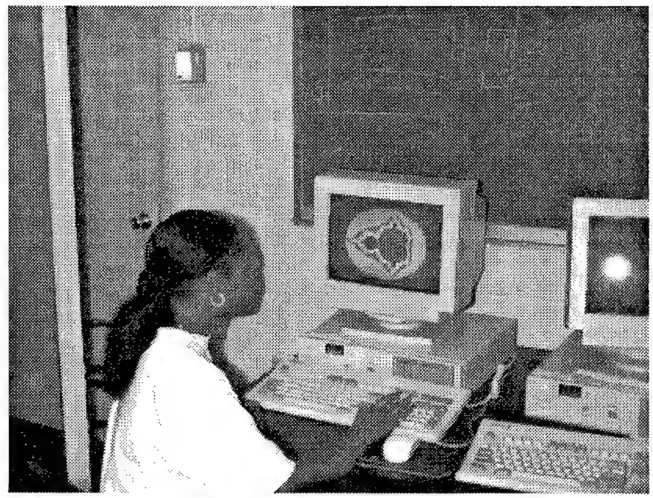
Summer of hard work!!

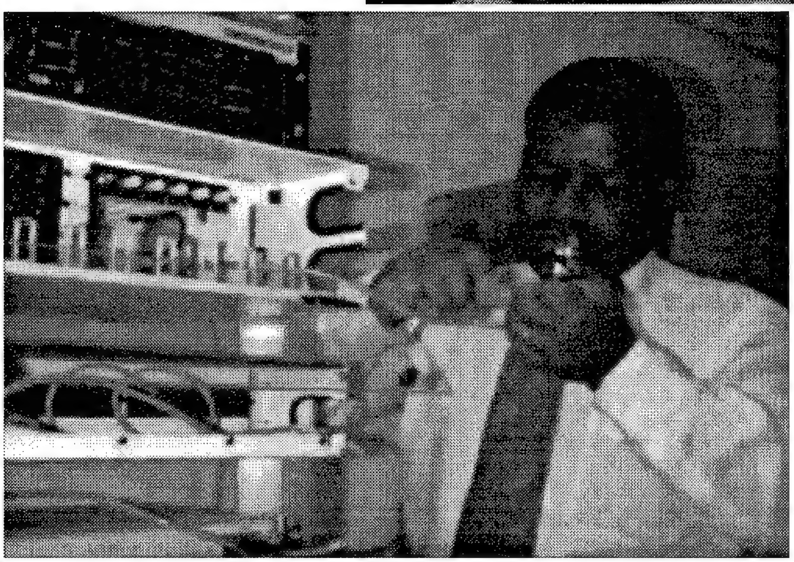
ATM is here!

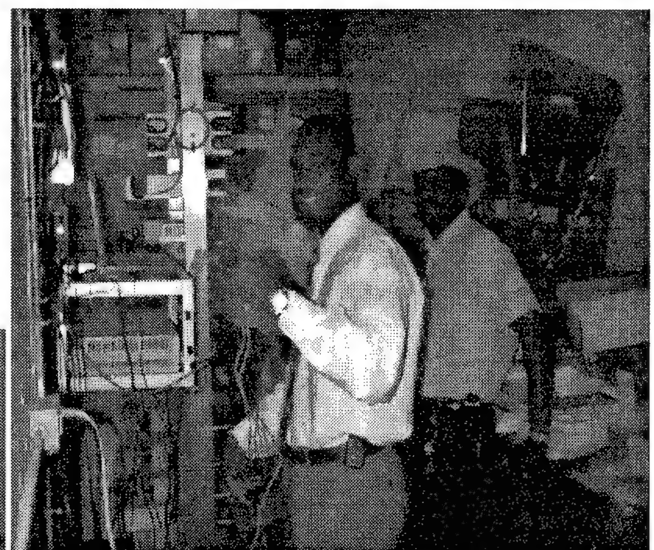


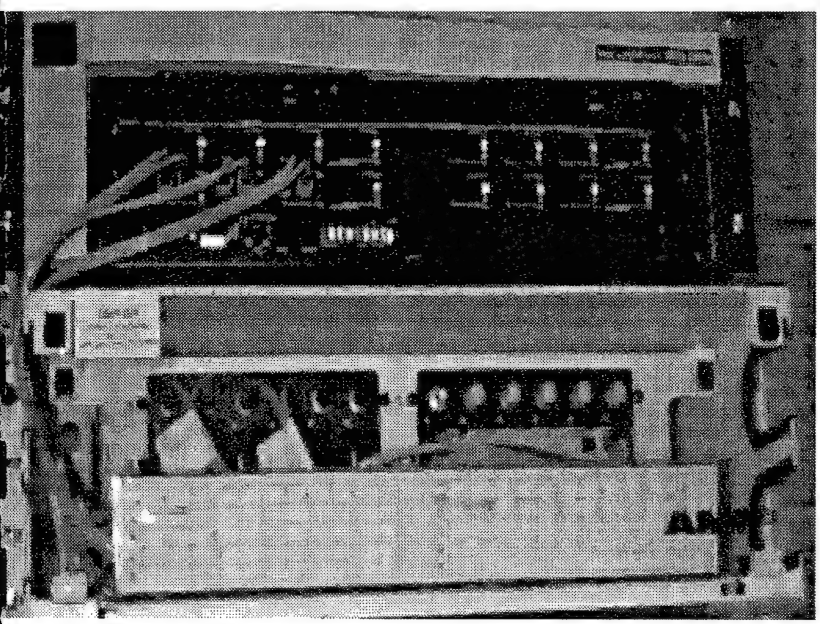












Nurturing ECSU Research Talent Program - ONR
Dr. Linda Hayden, PI
Box 672 ECSU
Elizabeth City, NC 27909
(919)335-3617
LHAYDEN@UMFORT.CS.ECSU.EDU
visit our Website ...
http://www.ecsu.edu

Fractals and Chaos With MATHEMATICA

Fractals and Chaos

Researched by :

Tanisha Cowell
Lakisha Mundon
Tammara Ward

Grad Student:

Timothy McCray

Mentor:

Dr. Manglik

Principle Investigator:

Dr. Linda Hayden

Final Report
Fractals and Chaos Team

Historical Developments

This week, the Fractals and Chaos group began our research by reviewing the first chapter of *Fractal Vision*, A History of Fractals and Chaos, surfing the Internet, working with the computer softwares, Mathematica, and Fractal Vision. We learned about great Mathematicians and philosophers such as Euclid of Alexandria, who "invented Geometry as we know it", or Rene Descartes, who "suggested that our universe could be measured by three intersecting perpendicular poles notched in perfectly even gradation, thus giving everything in existence a precise location in three straight-line dimensions. All of creation, then could be seen as a giant stack of tiny, perfectly cubic boxes." (Descartes' idea become the foundation for most of today's scientific views.) This novel approach to viewing the universe allowed people to perceive the space around them not as objects or events, but in abstract dimensions. Armed with the philosophy of Rene' Descartes, Sir Isaac Newton and Baron Gottfried Wilhelm von Leibnitz invented *differential calculus*. (The purpose of calculus is to turn the curved lines into linear ones. Ergo the equation dx/dt expresses the slope of an infinitesimally tiny line segment.) It was Leibnitz who proposed the idea that "all curves are made up of infinitesimally small line segments", also called tangent lines or differentials. (The only problem with this assumption is that curves resisted being entirely reduced to lines somehow.) From Leibnitz proposed claim, French astronomer Pierre-Simon Laplace voice the belief that " if the position and velocity of every particle in the universe was known, the curvilinear paths of every particle could be predicted with absolute certainty from simple linear equations." Then in the year 1875, a German mathematician Karl Weierstrass described a curve that couldn't be differentiated and therefore had no tangent lines. This caused chain of mathematical experiments to be performed. One example of these experiments is the Sierpinski's Triangle, which is also an example of a fractal. It is a triangle that has different numbers of stages. It starts with a

blank triangle and which is then divided into four equal pieces in the same likeness as the original triangle. This process is repeated over and over again ,or iterated, as the frequency of the triangle appears 3^n, and the area becomes$(3/4)^n$.(see appendix) The problem begins when the area of the covered region is to be found. Zero is never reached when finding the area.

Fractals

What then is a fractal? Fractals are rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced-size copy of the whole. Some examples of fractals are: Sierpinski's triangle, Cock's snowflake, Peano's curve, Mandelbrot set (example in appendix 1) and Lorenz attractor. Fractals are also used to describe clouds, mountains, turbulence, and coastlines, that do not correspond to simple geometric shapes. (It was Benoit Mandelbrot, who invented the word fractal from the Latin adjective fractus. The corresponding Latin verb, frangere, means "to break".)

Strange Attractor

A strange attractor is the limit set of a chaotic trajectory. A strange attractor is an attractor that is topologically distinct from periodic orbit or a limit cycle. A strange attractor can be considered a fractal attractor. Let us consider a volume in phase space defined by all the initial conditions a system may have. Far a dissipative system, this volume will shrink as the system evolves in time.(The Liouville's Theorem) If the system is sensitive to the initial conditions, trajectories of the points definite initial conditions will move apart in some directions, closer in others, but there will be a net shrinkage in volume. Ultimately, all points will lie along a fine line of zero volume. This is the strange attractor. All initial points in phase space which ultimately land on the attractor form a Basin of Attraction. A strange attractor results if a system is sensitive to initial conditions and is not conservative. While all chaotic attractors are strange, not all strange attractors are chaotic.

Mandelbrot Sets

Mandelbrot set is a fractal that is generated by a formal where the set of all complex c such that iterating $z \to z^2 + c$ does not go to infinity (starting with $z=0$). Zero is the critical point of $z^2 + c$. that is, a point where $d/dz\,(z^2 + c) = 0$. If you replace $z^2 + c$ with a different function, the starting value will have to e modified. For example, $z \to z^2 + z + c$, the critical point. Thus, testing the critical point shows if there is any stable attractive cycle. The difference between Mandelbrot set and Julia sets is simply Mandelbrot sets iterates z^2+c with z starting at 0 and varying c, and the Julia set iterates $z^2 +c$ for fixed c and varying starting z values. Meaning that the Mandelbrot set is in the parameter space(c-plane) while the Julia set exist in the dynamical or variable space(z-plane). The connection between the Mandelbrot set and the Julia sets are the point of c in the Mandelbrot set specifies the geometric structure of the corresponding Julia set.

It has been said that if a fractal is self-similar, you can specify mappings that map the whole onto the parts. Iteration of these mappings will conclude in convergence the of a fractal attractor. An Iterated function system consists of a collection of affine mappings. If a fractal can be describe by a diminutive number of mappings, the IFS is a very compact description of the fractal. Iterated function systems can be used to make things such as fractal ferns (appendix 2) and trees.

Linear Algebra through Mathematica

The Fractals and Chaos Research team has exploring **Mathematica**, a general software system for technical computations. The team adventured into the linear algebra (Eigenvalues and Eigenvectors) aspect of Mathematica. Our experimenting lead to the discovery that given an n*n matrix of real numbers, Mathematica will find the approximate numerical Eigenvalues and Eigenvectors. It also will give the characteristic polynomial.

In addition, Mathematica can calculate other functions related to linear algebra such as singular values, pseudo-Inverse matrices, and Jordan decomposition. Once our

knowledge of Mathematica was enhanced, we began our project with some affine transformation.

IFS and Affine transformation

An affine transformation of R^n is achieved by applying a linear transformation and following it with a translation

IFS 2,334,82

The Mathematics of IFS was developed by John Hutchinson and popularized by Michael Bainsley. IFS replaces polygons by other polygons as described by a generator. On every iteration, each polygon is replaces by a suitably scaled, rotated, and translated version of the polygons in the generator. It is also possible to derive a hopalong description which gives the image that would be created after iterating the geometric model to infinity. The description of this is a set of contractive transformations on a plane of the form

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

each with an assigned probability. To run the system an initial point is chosen and on each iteration one of the transformation is chosen randomly according to the assigned probabilities, the resulting points (xn, yn) are drawn.

The IFS approach provides a good frame work from which to pursue the mathematics of many classical fractals as well more general types. If is also the frame work from which to make the transition to chaos associated with fractals.

An affine transformation is one that scales time and distance by different factors.

For Example:

$T(u)=Au + y$

where A is a matrix and y is a fixed vector. An affine transformation can be interpreted as a matrix transformation followed by a translation.(see Appendix 3) Using affine transformation, we created Sierpinski's Triangle in both 2-D, and 3-D, as well as creating a checker board. (see Appendix 4,5,6)

Chaos

Chaos is apparently unpredictable behavior arising in a deterministic system because of great sensitivity to initial conditions. Chaos arises in a dynamical system if two arbitrarily close starting points diverge exponentially, so that their future behavior is eventually unpredictable. An example of chaos is the weather. It takes just a small variation of the initial conditions to result in radically different weather later.

Linear Algebra through Mathematica

The Fractals and Chaos Research team has exploring Mathematica, a general software system for technical computations. This week, the team adventured into the linear algebra (Eigenvalues and Eigenvectors) aspect of Mathematica. Our experimenting lead to the discovery that given an n*n matrix of real numbers, Mathematica will find the approximate numerical Eigenvalues and Eigenvectors. It also will give the characteristic polynomial.

In addition, Mathematica can calculate other functions related to linear algebra such as singular values, pseudo-Inverse matrices, and Jordan decomposition.

Fractal Vision: Fractals in the Real World

Through Fractal Vision, one is able to view a pictorial image of fractals. The team has been exploring fractals in the real world. In Fractal Vision, the team was able to see the progression of clouds (cirrus and stratus) by modeling the movements of air currents. By modeling the different types of air currents for each type of cloud, the software is able to approximate the shape of the cloud. The team also look at different types of trees (maple and pine) to explore their unique characteristic branching pattern, and furthermore, each leaf pattern. Throughout these experimentations, the team was able to get a better understanding of fractals in the real world.

Bibliography

Bourke, Paul. Dynamical Symmetries. http://www.wmin.ac.uk/~ccdva/

Bourke, Paul. The Math of Chaos. http://ddi.digital.net/~wisdom/fractal/chaosmth.html.

Bourke, Paul. IFS-Iterated Function Systems
http://www.auckland.ac.nz/arch/pdbourke/fractals/ifs.html.

Derbyshire, Paul. PGD's Quick Guide to the Mandelbrot set
http://chat.carlrton.ca/~pderbysh/manguide.html

Oliver, Dick. Fractal Vision. Sams Publishing, Indiana, 1992.

Williams, Gareth. Linear Algebra with application. Wm.C. Brown Publishers. Bogota, 1996.

Wolfram, Stephen. Mathematica: The Student Book. Addison-Wesley Publishing Company, Massachusetts. 1994.

APPENDIX 1

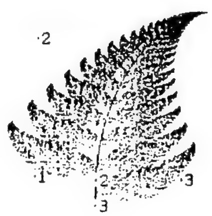
APPENDIX 2

Iterated Function Systems Playground

This page lets you design your own IFS fractal. For help how to operate it, please read the manual.

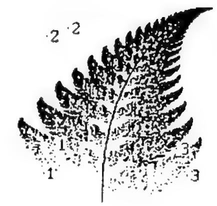
Transformations:

Transformation 1:



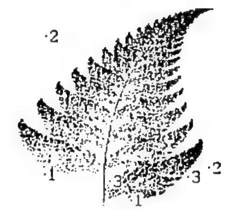
Weight =

Transformation 2:



Weight =

Transformation 3:



Weight =

Transformation 4:

Weight =

APPENDIX 3

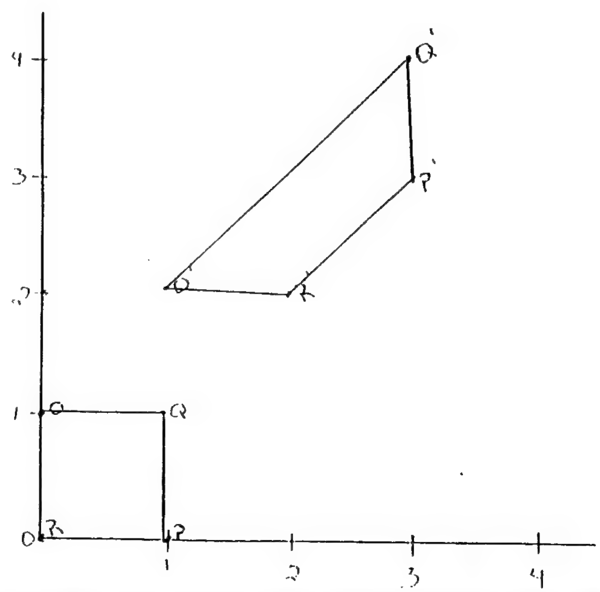
An **affine transformation** is a transformation of the form T: $R^n \to R$, defined by $T(u) = A u + v$
where A is a matrix and v is a fixed vector.

An affine transformation can be interpreted as a matrix transformation followed by a translation.

For example, consider the affine transformation on R^2.

$$T\begin{matrix}X\\Y\end{matrix} = \begin{matrix}2 & 1\\1 & 1\end{matrix}\ \begin{matrix}X\\Y\end{matrix} + \begin{matrix}1\\2\end{matrix}$$

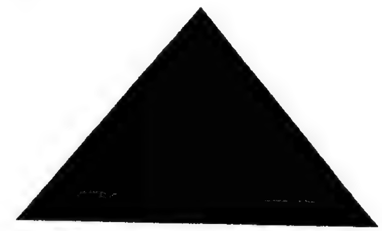
P	P'	Q	Q'	R	R'	O	O'
1	3	1	3	0	2	0	1
0	3	1	4	0	2	1	2



APPENDIX 4

Fracsum4

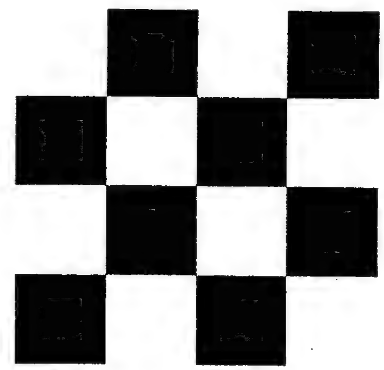
```
{RGBColor[1,0,0],Polygon[{{0.,0.},{1.,0.},{.5,1.}}]}
Show[Graphics[%]]
```



-Graphics-

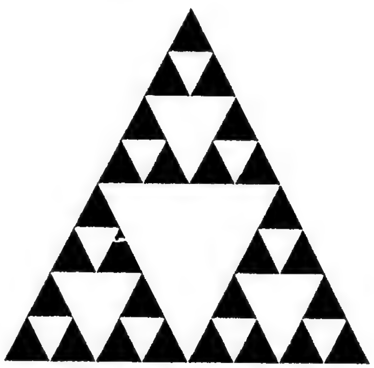
APPENDIX 5


```
Show[Graphics[
{{RGBColor[1,0,0],Rectangle[{0,1/2},{1/4,3/4}]},
{RGBColor[1,0,0],Rectangle[{0,0},{1/4,1/4}]},
{GrayLevel[.9],Rectangle[{0,1/4},{1/4,1/2}]},
{GrayLevel [.9],Rectangle[{0,3/4},{1/4,1}]},
{RGBColor[1,0,0],Rectangle[{1/4,3/4},{1/2,1}]},
{GrayLevel[.9],Rectangle[{1/4,1/2},{1/2,3/4}]},
{RGBColor[1,0,0],Rectangle[{1/4,1/4},{1/2,1/2}]},
{GrayLevel[.9],Rectangle[{1/4,0},{1/2,1/4}]},
{RGBColor[1,0,0],Rectangle[{1/2, 0},{3/4,1/4}]},
{GrayLevel [.9], Rectangle[{1/2,1/4},{3/4,1/2}]},
{RGBColor[1,0,0],Rectangle[{1/2,1/2},{3/4,3/4}]},
{GrayLevel [.9], Rectangle[{1/2,3/4},{3/4,1}]},
{RGBColor[1,0,0],Rectangle[{3/4,3/4},{1,1}]},
{GrayLevel [.9],Rectangle[{3/4,1/2},{1,3/4}]},
{RGBColor[1,0,0],Rectangle[{3/4,1/4},{1,1/2}]},
{GrayLevel [.9], Rectangle[{3/4,0},{1,1/4}]}}],
AspectRatio->Automatic]
```



-Graphics-


```
Show[Graphics[
{{RGBColor[1,0,0],Polygon[{{0,0},{1/2,0},{1/4,1/2}}],
RGBColor[1,0,0],Polygon[{{1/2,0},{1,0},{3/4,1/2}}]},
RGBColor[1,0,0], Polygon[{{1/4,1/2},{3/4,1/2},{1/2,1}}],
GrayLevel[1],Polygon[{{1/8,1/4},{1/4,0},{3/8,1/4}}],
GrayLevel[1],Polygon[{{5/8,1/4},{3/4,0},{7/8,1/4}}],
GrayLevel[1],Polygon[{{1/2,1/2},{3/8,3/4},{5/8,3/4}}],
GrayLevel[1],Polygon[{{1/16,1/8},{1/8,0},{3/16,1/8}}],
GrayLevel[1],Polygon[{{5/16,1/8},{3/8,0},{7/16,1/8}}],
GrayLevel[1],Polygon[{{3/16,3/8},{1/4,1/4},{5/16,3/8}}],
GrayLevel[1],Polygon[{{5/8,0},{9/16,1/8},{11/16,1/8}}],
GrayLevel[1],Polygon[{{13/16,1/8},{15/16,1/8},{7/8,0}}],
GrayLevel[1],Polygon[{{3/4,1/4},{13/16,3/8},{11/16,3/8}}],
GrayLevel[1],Polygon[{{5/16,5/8},{3/8,1/2},{7/16,5/8}}],
GrayLevel[1],Polygon[{{9/16,5/8},{11/16,5/8},{5/8,1/2}}],
GrayLevel[1],Polygon[{{1/2,3/4},{9/16,7/8},{7/16,7/8}}]}]
,AspectRatio->Automatic]
```

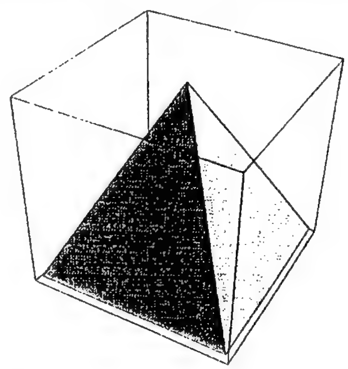


-Graphics-

```
Show[Graphics[%]]
```

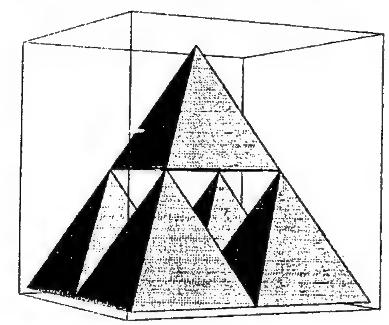
APPENDIX 6


```
Show[Graphics3D[{Polygon[{{0,0,0},{1,0,0},{1/2,1/2,1}}],
Polygon[{{1,0,0},{1,1,0},{1/2,1/2,1}}],
Polygon[{{0,1,0},{1,1,0},{1/2,1/2,1}}],
Polygon[{{0,1,0},{0,0,0},{1/2,1/2,1}}]}]]
```



-Graphics3D-


```
Show[Graphics3D[{
{Polygon[{{0,0,0},{1/2,0,0},{1/4,1/4,1/2}}],
Polygon[{{1/2,0,0},{1/2,1/2,0},{1/4,1/4,1/2}}],
Polygon[{{0,1/2,0},{1/2,1/2,0},{1/4,1/4,1/2}}],
Polygon[{{0,0,0},{0,1/2,0},{1/4,1/4,1/2}}]},
{Polygon[{{1/2,0,0},{1,0,0},{3/4,1/4,1/2}}],
Polygon[{{1,0,0},{1,1/2,0},{3/4,1/4,1/2}}],
Polygon[{{1/2,1/2,0},{1,1/2,0},{3/4,1/4,1/2}}],
Polygon[{{1/2,1/2,0},{1/2,0,0},{3/4,1/4,1/2}}]},
{Polygon[{{0,1/2,0},{1/2,1/2,0},{1/4,3/4,1/2}}],
Polygon[{{1/2,1,0},{1/2,1/2,0},{1/4,3/4,1/2}}],
Polygon[{{0,1,0},{1/2,1,0},{1/4,3/4,1/2}}],
Polygon[{{0,1,0},{0,1/2,0},{1/4,3/4,1/2}}]},
{Polygon[{{1/2,1/2,0},{1,1/2,0},{3/4,3/4,1/2}}],
Polygon[{{1,1,0},{1,1/2,0},{3/4,3/4,1/2}}],
Polygon[{{1/2,1,0},{1,1,0},{3/4,3/4,1/2}}],
Polygon[{{1/2,1,0},{1/2,1/2,0},{3/4,3/4,1/2}}]},
{Polygon[{{1/4,1/4,1/2},{3/4,1/4,1/2},{1/2,1/2,1}}],
Polygon[{{3/4,1/4,1/2},{3/4,3/4,1/2},{1/2,1/2,1}}],
Polygon[{{1/4,3/4,1/2},{3/4,3/4,1/2},{1/2,1/2,1}}],
Polygon[{{1/4,3/4,1/2},{1/4,1/4,1/2},{1/2,1/2,1}}]}}],
ViewPoint->{4.000,-2.112,-0.060}]
```



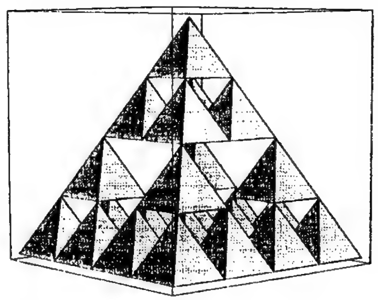
-Graphics3D-

```
Show[Graphics3D[{
{Polygon[{{1/4,0,0},{0,0,0},{1/8,1/8,1/4}}],
Polygon[{{1/4,0,0},{1/4,1/4,0},{1/8,1/8,1/4}}],
Polygon[{{0,1/4,0},{1/4,1/4,0},{1/8,1/8,1/4}}],
Polygon[{{0,0,0},{0,1/4,0},{1/8,1/8,1/4}}]},
{Polygon[{{1/4,0,0},{1/2,0,0},{3/8,1/8,1/4}}],
Polygon[{{1/2,0,0},{1/2,1/4,0},{3/8,1/8,1/4}}],
Polygon[{{1/4,1/4,0},{1/2,1/4,0},{3/8,1/8,1/4}}],
Polygon[{{1/4,0,0},{1/4,1/4,0},{3/8,1/8,1/4}}]},
{Polygon[{{1/2,0,0},{3/4,0,0},{5/8,1/8,1/4}}],
Polygon[{{3/4,1/4,0},{3/4,0,0},{5/8,1/8,1/4}}],
Polygon[{{1/2,1/4,0},{3/4,1/4,0},{5/8,1/8,1/4}}],
Polygon[{{1/2,0,0},{1/2,1/4,0},{5/8,1/8,1/4}}]},
{Polygon[{{3/4,0,0},{1,0,0},{7/8,1/8,1/4}}],
Polygon[{{1,1/4,0},{1,0,0},{7/8,1/8,1/4}}],
Polygon[{{3/4,1/4,0},{3/4,0,0},{7/8,1/8,1/4}}],
Polygon[{{3/4,1/4,0},{1,1/4,0},{7/8,1/8,1/4}}]},
{Polygon[{{0,1/4,0},{1/4,1/4,0},{1/8,3/8,1/4}}],
Polygon[{{1/4,1/2,0},{1/4,1/4,0},{1/8,3/8,1/4}}],
Polygon[{{0,1/2,0},{1/4,1/4,0},{1/8,3/8,1/4}}],
Polygon[{{0,1/2,0},{0,1/4,0},{1/8,3/8,1/4}}]},
{Polygon[{{1/4,1/4,0},{1/2,1/4,0},{3/8,3/8,1/4}}],
Polygon[{{1/2,1/2,0},{1/2,1/4,0},{3/8,3/8,1/4}}],
Polygon[{{1/4,1/2,0},{1/2,1/2,0},{3/8,3/8,1/4}}],
Polygon[{{1/4,1/2,0},{1/4,1/4,0},{3/8,3/8,1/4}}]},
{Polygon[{{1/2,1/4,0},{3/4,1/4,0},{5/8,3/8,1/4}}],
Polygon[{{3/4,1/2,0},{3/4,1/4,0},{5/8,3/8,1/4}}],
Polygon[{{1/2,1/2,0},{3/4,1/2,0},{5/8,3/8,1/4}}],
Polygon[{{1/2,1/4,0},{1/2,1/2,0},{5/8,3/8,1/4}}]},
{Polygon[{{3/4,1/4,0},{1,1/4,0},{7/8,3/8,1/4}}],
Polygon[{{1,1/2,0},{1,1/4,0},{7/8,3/8,1/4}}],
Polygon[{{3/4,1/2,0},{1,1/2,0},{7/8,3/8,1/4}}],
Polygon[{{3/4,1/4,0},{3/4,1/2,0},{7/8,3/8,1/4}}]},
{Polygon[{{0,1/2,0},{1/4,1/2,0},{1/8,5/8,1/4}}],
Polygon[{{0,1/2,0},{0,3/4,0},{1/8,5/8,1/4}}],
Polygon[{{1/4,3/4,0},{1/4,1/2,0},{1/8,5/8,1/4}}],
Polygon[{{0,3/4,0},{1/4,3/4,0},{1/8,5/8,1/4}}]},
{Polygon[{{1/4,1/2,0},{1/2,1/2,0},{3/8,5/8,1/4}}],
Polygon[{{1/2,1/2,0},{1/2,3/4,0},{3/8,5/8,1/4}}],
Polygon[{{1/4,3/4,0},{1/2,3/4,0},{3/8,5/8,1/4}}],
Polygon[{{1/4,1/2,0},{1/4,3/4,0},{3/8,5/8,1/4}}]},
{Polygon[{{1/2,1/2,0},{3/4,1/2,0},{5/8,5/8,1/4}}],
Polygon[{{3/4,1/2,0},{3/4,3/4,0},{5/8,5/8,1/4}}],
Polygon[{{1/2,3/4,0},{3/4,3/4,0},{5/8,5/8,1/4}}]
Polygon[{{1/2,1/2,0},{1/2,3/4,0},{7/8,5/8,1/4}}]},
{Polygon[{{3/4,1/2,0},{1,1/2,0},{7/8,5/8,1/4}}],
Polygon[{{1,1/2,0},{1,3/4,0},{7/8,5/8,1/4}}],
Polygon[{{3/4,3/4,0},{1,3/4,0},{7/8,5/8,1/4}}],
```



```
Polygon[{{3/4,1/2,0},{3/4,3/4,0},{7/8,5/8,1/4}}]},
{Polygon[{{0,3/4,0},{1/4,3/4,0},{1/8,7/8,1/4}}],
Polygon[{{1/4,3/4,0},{1/4,1,0},{1/8,7/8,1/4}}],
Polygon[{{0,1,0},{1/4,1,0},{1/8,7/8,1/4}}],
Polygon[{{0,3/4,0},{0,1,0},{1/8,7/8,1/4}}]},
{Polygon[{{1/4,3/4,0},{1/2,3/4,0},{3/8,7/8,1/4}}],
Polygon[{{1/2,1,0},{1/2,3/4,0},{3/8,7/8,1/4}}],
Polygon[{{1/4,1,0},{1/2,1,0},{3/8,7/8,1/4}}],
Polygon[{{1/4,3/4,0},{1/4,1,0},{3/8,7/8,1/4}}]},
{Polygon[{{1/2,3/4,0},{3/4,3/4,0},{5/8,7/8,1/4}}],
Polygon[{{3/4,1,0},{3/4,3/4,0},{5/8,7/8,1/4}}],
Polygon[{{1/2,1,0},{3/4,1,0},{5/8,7/8,1/4}}],
Polygon[{{1/2,3/4,0},{1/2,1,0},{5/8,7/8,1/4}}]},
{Polygon[{{3/4,3/4,0},{1,3/4,0},{7/8,7/8,1/4}}],
Polygon[{{1,3/4,0},{1,1,0},{7/8,7/8,1/4}}],
Polygon[{{3/4,1,0},{1,1,0},{7/8,7/8,1/4}}],
Polygon[{{3/4,1,0},{3/4,3/4,0},{7/8,7/8,1/4}}]},
{Polygon[{{1/8,1/8,1/4},{3/8,1/8,1/4},{1/4,1/4,1/2}}],
Polygon[{{3/8,1/8,1/4},{3/8,3/8,1/4},{1/4,1/4,1/2}}],
Polygon[{{1/8,3/8,1/4},{3/8,3/8,1/4},{1/4,1/4,1/2}}],
Polygon[{{1/8,1/8,1/4},{1/8,3/8,1/4},{1/4,1/4,1/2}}]},
{Polygon[{{5/8,1/8,1/4},{7/8,1/8,1/4},{3/4,1/4,1/2}}],
Polygon[{{7/8,1/8,1/4},{7/8,3/8,1/4},{3/4,1/4,1/2}}],
Polygon[{{5/8,3/8,1/4},{7/8,3/8,1/4},{3/4,1/4,1/2}}],
Polygon[{{5/8,1/8,1/4},{5/8,3/8,1/4},{3/4,1/4,1/2}}]},
{Polygon[{{1/8,5/8,1/4},{3/8,5/8,1/4},{1/4,3/4,1/2}}],
Polygon[{{3/8,5/8,1/4},{3/8,7/8,1/4},{1/4,3/4,1/2}}],
Polygon[{{1/8,7/8,1/4},{3/8,7/8,1/4},{1/4,3/4,1/2}}],
Polygon[{{1/8,5/8,1/4},{1/8,7/8,1/4},{1/4,3/4,1/2}}]},
{Polygon[{{5/8,7/8,1/4},{7/8,7/8,1/4},{3/4,3/4,1/2}}],
Polygon[{{5/8,5/8,1/4},{7/8,5/8,1/4},{3/4,3/4,1/2}}],
Polygon[{{7/8,5/8,1/4},{7/8,7/8,1/4},{3/4,3/4,1/2}}],
Polygon[{{5/8,5/8,1/4},{5/8,7/8,1/4},{3/4,3/4,1/2}}]},
{Polygon[{{1/4,1/4,1/2},{1/2,1/4,1/2},{3/8,3/8,3/4}}],
Polygon[{{1/2,1/4,1/2},{1/2,1/2,1/2},{3/8,3/8,3/4}}],
Polygon[{{1/4,1/2,1/2},{1/4,1/4,1/2},{3/8,3/8,3/4}}],
Polygon[{{1/4,1/2,1/2},{1/2,1/2,1/2},{3/8,3/8,3/4}}]},
{Polygon[{{1/2,1/4,1/2},{3/4,1/4,1/2},{5/8,3/8,3/4}}],
Polygon[{{3/4,1/2,1/2},{3/4,1/4,1/2},{5/8,3/8,3/4}}],
Polygon[{{1/2,1/2,1/2},{3/4,1/2,1/2},{5/8,3/8,3/4}}],
Polygon[{{1/2,1/4,1/2},{1/2,1/2,1/2},{5/8,3/8,3/4}}]},
{Polygon[{{1/4,1/2,1/2},{1/2,1/2,1/2},{3/8,5/8,3/4}}],
Polygon[{{1/2,3/4,1/2},{1/2,1/2,1/2},{3/8,5/8,3/4}}],
Polygon[{{1/4,3/4,1/2},{1/2,3/4,1/2},{3/8,5/8,3/4}}],
Polygon[{{1/4,1/2,1/2},{1/4,3/4,1/2},{3/8,5/8,3/4}}]},
{Polygon[{{1/2,1/2,1/2},{3/4,1/2,1/2},{5/8,5/8,3/4}}],
Polygon[{{3/4,3/4,1/2},{3/4,1/2,1/2},{5/8,5/8,3/4}}],
Polygon[{{1/2,3/4,1/2},{3/4,3/4,1/2},{5/8,5/8,3/4}}],
```

```
Polygon[{{1/2,1/2,1/2},{1/2,3/4,1/2},{5/8,5/8,3/4}}]},
{Polygon[{{3/8,3/8,3/4},{5/8,3/8,3/4},{1/2,1/2,1}}],
Polygon[{{5/8,5/8,3/4},{5/8,3/8,3/4},{1/2,1/2,1}}],
Polygon[{{3/8,5/8,3/4},{5/8,5/8,3/4},{1/2,1/2,1}}],
Polygon[{{3/8,3/8,3/4},{3/8,5/8,3/4},{1/2,1/2,1}}]}}}],
ViewPoint->{3.950,-3.355,0.398}]
```



```
-Graphics3D-
```

HTML/JAVA

HTML/JAVA Team
Final Report
August 2, 1996

Courtney Fields
Makeba Fussell
Kuchumbi Hayden
Reginald Turner
Kimberly Wright
Michelle Brown, Graduate Student
Marie Dail, Graduate Student
Tracy Chamberlain, Mentor

Outline

- HTML Techniques
  - Tables
  - Frames
  - Animated Gifs
  - Java
- ECSU Homepage

Tables

- Before tags for tables were finalized it was necessary to use the <pre> tag for tabular information.
- Tables are very useful for the presentation of tabular information.
- They are also excellent means of presenting regular information for creative HTML authors.

Table Elements

The general format of a table looks like this:

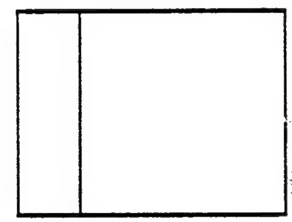
<TABLE> - start of table definition
<CAPTION> caption contents </CAPTION> - caption definition
<TR> - start of first row definition
<TH> cell contents </TH> - first cell in row 1 (a head)
<TH> cell contents </TH> - last cell in row 1 (a head)
</TR> - end of first row definition
<TR> - start of second row definition
<TD> cell contents </TD> - first cell in row 2
<TD> cell contents </TD> - last cell in row 2
</TR> - end of second row definition
<TR> - start of last row definition
<TD> cell contents </TD> - first cell in last row
<TD> cell contents </TD> - last cell in last row
</TR> - end of last row definition
</TABLE> - end of table definition

Frames

- Divide web pages into multiple, scrollable regions.
- Each frame has several features
  - an individual URL
  - given a NAME
  - resize if the user changes the window's size.
- Elements that the user should always see can be placed in a static individual frame.

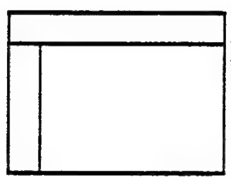
Frames Syntax

```
<frameset cols="30%,70%">
<framesrc="contents.html">
<framesrc="linkone.html"name="MAIN">
</frameset>
```



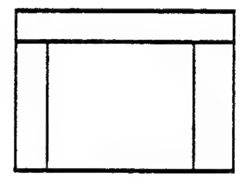
Frames - Examples

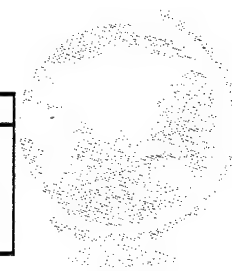
```
<frameset rows="25%,*">
<framesrc="linktwo.html" name="banner"  scrolling="yes">
   <frameset cols="30%,70%">
   <frame src="contents.html">
   <frame src="linkthree.html" name=  "main">
   </frameset>
</frameset>
```



Frames - Examples

```
<frameset rows="25%,*">
<framesrc="linkone.html" name="banner" >
   <frameset cols="25%,50%,25%">
   <frame src="jordandunk.html">
   <frame src="shaqdunk3.jpg>
   <frame src="kempdunk.jpg">
   </frameset>
</frameset>
```





Animated GIFS

- Animated GIFS are called GIF89a images.
- Most GIFs over the years have only one image per file.
- Most programs that work with GIF are designed around the idea of one image per file.

Animated GIFS

- GIF89a allows multiple images to be compiled within a single GIF file.
- Single GIF file you reference in your HTML pages will display multiple images, in sequence, just like flip-book animation.

Animated GIFS

- GIF animations are showing up everywhere.
- Animated GIFS are created by individuals in their spare time and are free.
- Everyone is finding merit in their implementation and fun in their use.

Creating Animated GIFS

Nine steps to animation using GifBuilder for Macs:

- Pick the image that you wish to animate.
- Make the image rotate in the style you wish the animation to appear. (Hint: alphabetically title each picture.)
- Put images on the desktop.
- Using GifBuilder insert images into frames.
- Arrange images correctly.
- Make your specifications.
- Click on Run icon and select start to view your progress.
- Copy animated image to the correct directory.
- Place the image into the html document using normal html formats.

ECSU Homepage

- Working with University Relations Office in designing the webpage.
- Provided us with an outline of how the page should look.
- They are providing us with the information that needs to be typed, scanned, formatted.
- Students are typing in catalogs, handbooks, brochures on word processors then using ftp they put the files into the account on the server which is housing all ECSU webpage information.

ECSU Homepage

- Once the files are in the account the files are then coded into HTML formats, backgrounds, icons, gifs, bullets, bars, etc. are added to complete the page.
- Once completed University Relations will then come to view the page and be given a printout of the page to be given to the appropriate department for proofing.
- If changes are needed then University Relations will return the pages with corrections to us and the changes are made.

Elizabeth City State University

About ECSU

Admissions Information

Academics & Research

Libraries

Student Life

Athletics

Administrative

Alumni, Development & Planning

About ECSU

- Introduction
- History of the University of North Carolina
    - ECSU Mission
    - Campus Map
- Degrees Available
- News
- Directory

Elizabeth City State University

Dismal Swamp Boardwalk Project

Development and Purpose

The Dismal Swamp Boardwalk Project was completed and dedicated by Elizabeth City State University in the Spring of 1994. The wetlands property, consisting of 639 acres, was acquired by the University from the Department of Health, Education and Welfare. The half-mile long boardwalk and observation tower were constructed with Title III funds, and its primary function is to provide access to a wetlands wilderness area for use in research and educational activities.

Elizabeth City State University
Department of
Mathematics & Computer Science

- NASA-NRTS at ECSU-(Regional Training Site)
- ONR Nurturing ECSU Research Talent-(NERT) Program
- CS Student Homepages

Scholarship Opportunities

- ECSU- ONR Scholarship Program
- NASA Regional Network and Training Center Scholarship Program

NASA-NRTS Service Award Winners

UNDER CONSTRUCTION

Welcome to the

Nurturing ECSU Research Talent-(NERT) Program

Funded by the Office of Naval Research

The Office of Naval Research (ONR) coordinates, executes, and promotes the science and technology programs of the United States Navy and Marine Corps through universities, government laboratories, and nonprofit organizations. It provides technical advice to the Chief of Naval Operations and the Secretary of the Navy, works with industry to improve technology manufacturing processes while reducing fleet costs, and fosters continuing academic interest in naval relevant science from the high school through post-doctoral levels.

Research Teams

- Multimedia Authoring
- Fractals and Chaos
- Computer Graphics
- Unix System Administration
- Mott Scattering
- Statistical Analysis
- Numerical Analysis

Summer '95 Research Project

Conference Reports

Elizabeth City State University

Music Department

Music Industry Studies

Within the Music Industry Studies Degree Program, concentrations are offered in Music Business Administration and Music Engineering & Technology.

The Music Business Administration concentration focuses on music business, management, marketing, sales, publishing, retailing, and promotion. The Music Engineering & Technology concentration is based on state-of-the-art, 24-track recording and MIDI/electronic music studios. The curriculum incorporates studies in audio, MIDI, and computer applications.

The Music Industry Studies Degree Program provides students with the opportunity to record, produce, and market actual products through the student operated record label, music publishing, and music production companies.

DEGREES OFFERED

Bachelor of Science in Music Industry Studies
Bachelor of Arts in Music

CONCENTRATIONS

Music Engineering & Technology
Music Business Administration
Voice Theory & Composition
Piano & Organ
Brass
Woodwinds
Percussion

PERFORMING GROUPS

1. Concert Band
2. Marching Band

   Collegians Jazz Ensemble

3. Brass Ensemble
4. Woodwind Ensemble
5. Percussion Ensemble
6. University Choir
7. Choral Ensemble
8. Vocal Jazz Ensemble
9. Gospel Choir
10. Collegium Musicum

To return to the ECSU Homepage, click here

Making waves on the WWW

By Steve Henning

Before you dive into the waters of the World Wide Web, there are a few things to consider that can help you keep your head above water.

The physical installation of a Web server may be the least of your worries because the way your information is presented is important in this graphics-based medium.

Originating in Switzerland at the European Particle Physics Laboratory (CERN), the World Wide Web was designed as a means of sharing information and resources over the Internet. The Web, as it is called by Net surfers, is actually a network of servers within the Internet that provides information to users in the form of interactive documents.

These documents may reside on the same server or on any other server on the Web. A user can view a document on a server in Illinois and, with the click of the mouse, immediately bring up another that resides on a server in France.

Today, many organizations are setting up their own Web sites to provide information to customers online. Whether it's worth the time and money to create a Web site often depends on how well the Web developer takes advantage of the unique capabilities of the technology.

A Web server can reside on a number of platforms, including Unix, Windows/NT and Macintosh. Web "daemon" software is used to set up Web servers and is available from various sources via the Web and ftp. For Unix and other platforms, the CERN HTTPD (hypertext transfer protocol daemon) software can be accessed via the Web at http://info.cern.ch/hypertext/WWW/Daemon/Status.html

Another issue to consider is the type of connection the machine will have to the Internet. The bandwidth of the line you choose will depend on the amount of traffic you expect on your Web site.

There are also many Internet providers who will charge for space on their Web servers, letting you provide the content, while they handle all the server hardware and connection issues. Detailed information on the nuts and bolts of setting up a Web server can also be found at the above-mentioned location. Because the physical setup information is readily available, this article focuses on presentation issues, which are often overlooked.

How are the documents for the Web created? They are scripted using the Hypertext Markup Language (HTML). HTML, which is a subset of the Standard

See WWW, page 44

Copyright compliance programs protect against software piracy

In October 1994, Ultea Enterprises, a Detroit-based auto parts manufacturer, paid \$200,000 to settle a software piracy lawsuit.

Only three months later, Louis Berger, a New Jersey-based international architectural and engineering consulting firm, settled for \$200,000 following a software piracy investigation. In addition to the settlement amounts, each organization had to purchase legal software, incur legal expenses and suffer potential damage to its reputation among customers, employees and vendors.

The lawsuits against these civil engineering firms resulted from tips to the Business Software Alliance's (BSA) anti-piracy hotline (800-688-2721). BSA is a Washington, D.C.-based trade organization dedicated to fighting software piracy. The trade group counts as its members Autodesk, Bentley Systems, Intergraph, Lotus Development Corp., Microsoft, Novell, WordPerfect Applications Group and The Santa Cruz Operation.

U.S. law prohibits duplicating software for profit, making multiple copies for use by different users within an organization and giving an unauthorized copy to another individual. If caught pirating software, an individual or organization may be tried under civil and criminal law.

A civil action may result in statutory damages of as much as \$100,000 per infringement. Criminal penalties include fines of as much as \$250,000 and jail terms of as long as five years, or both.

Despite these laws, piracy costs the software industry an estimated \$12.8 billion each year, according to BSA officials.

Software presents a unique problem because it is easy to duplicate, and the copy is often indistinguishable from the original. According to Sandra Boulton, Autodesk's director of anti-piracy, "Unlike other works, such as audio and video tapes, there is no degeneration in quality from copy to copy, and the one hundredth or one thousandth copy can be a perfect replication of the original."

A program that reflects years of effort by a team of software developers, and millions of dollars of investment, takes only a few minutes to copy. Although software is expensive to develop, duplication costs little or nothing; virtually any PC can be used to make unauthorized copies.

Software theft takes many forms, and the reasons for unauthorized software copying range from pure profit motive to carelessness, ignorance of the law and a general disregard for the importance of treating software as valuable intellectual property.

"Software piracy stifles innovation, destroys the financial incentives for the creation of new programs and applications, and menaces the continued growth of the industry," Boulton said.

According to BSA's director of

See PIRACY, page 43

WWW

Generalized Markup Language (SGML), is made up of a set of commands that define how a Web document is structured, including styles for titles, subheads and body text, as well as graphics images and hypertext links.

Although these styles define relative font sizes (a title being larger than a subtitle, for example), HTML does not define formatting. Instead, text formatting, including font and size of text, is defined from within the Web browser software itself (see below).

This can pose a challenge to Web page authors because they lack control over how the pages are displayed to the individual user. One way around this is to create bitmap graphics containing text if you want certain text, such as titles and logos, to be viewed in a specific format.

There are a few tools that ease the burden of Web page authoring. SoftQuad's HotMetal Pro, Internet Assistant, a free add-on to Microsoft Word, and Cyberleaf by Interleaf, will convert Word, WordPerfect and FrameMaker files to HTML documents.

The HTML documents, or Web pages, are viewed and navigated using Web browser software that resides on the local computer, the client to the Web servers. Well-known browsers include Enhanced Mosaic by Spyglass, Netscape by Netscape Communications and NCSA Mosaic.

Because there are different versions of HTML, it is important to test the functionality on as many Web browsers as possible when developing Web documents.

Companies who wish to use the Web should also consider that many people using the Web use limited text-only browsers such as Lynx, which is unable to download graphics images to the screen.

There are also alternatives to Web browsers that offer greater control over document formatting.

Adobe Acrobat, for instance, will read files in the Portable Document Format (PDF), which is an export option from many PostScript-based page layout and design applications, such as QuarkXpress. Using Acrobat, documents can be viewed as they appear in their original form.

Graphics viewed on the Web are most often formatted as gif files. Because these graphics are seen only on a computer screen, they need only be at a 72 dpi resolution.

Also, because the file size of the graphics should be as small as possible to accommodate rapid data transfer over limited bandwidth, the color palette needs to be reduced to as few colors as possible. One way to determine the optimal size for a graphics file is to test the speed at which gif files of various sizes are downloaded to the client computer using the lowest bandwidth connection that a typical user would have.

A Web server is an informational tool. The format should be simple and easily navigated. If users find your Web difficult and unwieldy to explore, they'll give up. The challenge of a good Web site is to take advantage of the interactive nature. Graphics should be used both to help illustrate information and to help define its structure.

The flow of information on the Web is not one directional. A Web server can be used for the collection of user information. HTML forms easily can be set up to gather customer feedback, for instance. Then the collected data can be sent to an e-mail account or transferred by another application to a database.

Web documents can also be used without a Web. Organizations can take advantage of HTML documents and browsers for the internal distribution of information, too.

Advance planning is critical to taking full advantage of the Web. If Web technology is the next "wave," then resolving these issues early can help you make a bigger splash.

For information, call 206 217 7391. ■

Steve Henning is a member of the Walter Rubin & Onno Web server team.

Publishing on the World Wide Web: Organization and Design

By Patrick J. Lynch

Patrick J. Lynch, director of Yale University's Center for Advanced Instructional Media, considers the organizational and technical implications of publishing on the World Wide Web, as well as the creation of an effective interface to electronically published materials via well designed Web pages.

Every graphic designer or editor who has ever delivered the final originals of a document to a printer knows a very special kind of fear: it's the fear that you are just about to pay someone a great deal of money to produce 10,000 copies of some mistake you made and didn't catch, despite all your proofreading. Publishing is not for the careless or faint of heart, especially in academic organizations where disseminating information is at the heart of the enterprise. The logistics and expense of moving information from your desk to a wide audience are always formidable when paper is involved, and few organizations have more information-related expenses per employee than colleges and universities.

For years now we've been hearing about the advantages of electronic publishing over data networks, where just one copy of a document exists and everyone can access it via the network. The logic is compelling: Find a typo? Want to update a phone number? All you have to do is change the digital "original" and everyone on the network now has a new "copy" to read. Instead of dead information embalmed on paper, we'll have live, up-to-the-minute information, instantly available. But the rhetoric of electronic publishing has rarely matched the reality, and between technical networking headaches, awkward authoring tools, and cross-platform incompatibilities, the progress of networked electronic publishing just stumbled along—until about a year ago.

Beyond Print, but Not Quite Paperless

If you have been paying attention to the computer press lately you have probably heard about the World Wide Web (WWW) and graphical Internet browsers like NCSA's Mosaic and Netscape Corporation's Netscape Navigator. (If you've somehow missed all the fuss over the Internet's first "killer app," see Ralph Abraham's introduction to the WWW in February's issue of *Syllabus*.) Publishing text, pictures, sound, and even video clips over the Internet has never been easier, and the number of Web users running the Internet is astonishing. At Yale's Center for Advanced Instructional Media, we put up a WWW site in mid-January of this year (http://info.med.yale.edu/caim/C_HOME.HTML). By mid-February the site had been "hit" by individual WWW users almost 400 times, and that was before we even had a chance to publicize it in the usual WWW indexes and catalogs. The *New York Times* recently quoted an estimate that within a year from now, up to half of all Internet traffic may be related to World Wide Web documents. Current estimates of WWW users range up to 20 million, and with the recent linking of Prodigy, CompuServe, and America Online to the WWW, the number of potential new users continues to grow rapidly. Today there are few more cost-effective ways to disseminate information than through WWW documents delivered over the Internet.

Before you brace yourself for another starry-eyed paean to the "paperless office of the future" let me admit that most academics working today will probably print anything they're really interested in from the Web, and file it away with the rest of their reprints. Paper is comforting and familiar, and those of us who have grown up depending on it are loathe to give it up even when we fully understand the advantages of on-line documents. But I suspect that the next generation of academics will be much less attached to paper, and more dependent on (and more demanding of) electronic documents and information networks. Paper will never completely go away, but the trend lines for growth in paper publications will flatten over the next decade as more and more reference information and professional communication goes digital. The economics of publishing make this inevitable, and academic publications and reference works will lead the way.

Don't believe me? Look what has happened to encyclopedias: sales of the digital CD-ROM versions have surpassed paper versions this year, and at the current rate, there may not be any paper encyclopedias in production two years from now (collectors take note). The cost advantages of Internet publishing or publishing on CD-ROM are so great that the capital-starved, price-sensitive world of academic books and professional journal publishing will become primarily digital and networked long before the mainstream publishing giants convert most of their backlists to digital formats.

Educational Web Publishing: Not Just Another Pretty Interface

The implications of WWW electronic publishing by educational institutions fall into two interrelated information management categories: the transmission of information to faculty, staff, and students within the organization (the internal agenda); and what is intended for the rest of the world, including academic colleagues, prospective students, alumni, and prospective donors (the external agenda). Addressing both agendas will probably require some fundamental changes in your school's administrative and academic information management policies, starting with a realization that your school's WWW pages and any other Internet-accessible information you have posted may already be one of the most widely seen and influential views the world has of your campus and institutional behavior. Is anyone on your campus asking just what your WWW pages are saying about your university? They will.

The WWW is so new and has grown so fast that most universities have not had a chance to review and make formal policy decisions with respect to how their information is posted in WWW pages, what editorial and design standards should be used, and how to better coordinate and link all the bits and pieces of information that are already posted on their university file servers. Most of the WWW activity in universities has grown informally over the last year, mostly as a result of the grass-roots efforts of individual faculty, staff, and students. On most campuses this has resulted in a heterogeneous mix of styles, messages, and quality levels that are haphazardly linked together into campus WWW "home pages." Without an organized campus effort aimed at harnessing the power and capabilities of the WWW, much of the potential usefulness of the medium will be lost in a chaotic tangle that is neither easy to use, nor stable enough to depend on for important academic and administrative information. The challenge is to begin to coordinate and harmonize the "look and feel" of your university's Internet presence without quashing the creativity and enthusiasm that makes the WWW such an interesting vehicle for information publishing.

The first task is to recognize that your university needs a consistent, coordinated approach to the electronic publishing of information. The next task is to define what you want to say, who you want to say it to, and how you will organize those efforts to present the same professional, high-quality standards of content and production values you would insist on for any printed communication from your university. Carefully designed WWW pages are not just a matter of setting the right stylistic "tone" in your internal and external communications. Properly designed WWW sites, with coordinated graphic design, high editorial standards, and consistent user interface conventions, are the only way to insure that your investment in WWW information publishing will pay off by successfully convincing your various audiences that your WWW publications are the quickest, easiest, and most reliable places to find university information.

World Wide Web: Whence, Whither, What Next?

The author presents a snapshot of the World Wide Web after about half a decade, and speculates about where this young medium might be improved and which directions it might take from a technical perspective.

Henning Schulzrinne

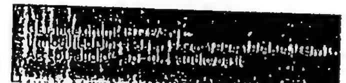
In a time span of about five years, the World Wide Web (WWW) [1] has become, next to electronic mail, the most popular Internet application. It has been a major contributor in turning the Internet, once an obscure data network for scientists and computer programmers, into a household word. The WWW allows users to retrieve text and multimedia objects from servers located throughout the world, with objects connected by hypermedia links. This article aims to provide a snapshot of the Web after about half a decade, speculating at the same time where this young medium might be improved and which directions it might take from a technical perspective.

Like most (successful) Internet technologies, the underlying central functionality of the Web is rather simple: a naming mechanism for files (the universal resource locator, URL), a typed, stateless retrieval protocol (Hypertext Transfer Protocol — HTTP), and a minimal formatting language with hyperlinks (hypertext markup language — HTML). Building a minimal "Web retriever" in a string-oriented language like Tcl or Perl requires a few dozen lines of code, with even less effort to build a server. Indeed, simply telneting to port 80 and issuing a command like GET /index.html is sufficient to retrieve documents. All of the basic technologies were around prior to the "invention" of the Web, generally credited to Tim Berners-Lee and Robert Cailliau at CERN around 1989.[1] However, the major accomplishment was not an individual protocol, but rather the integration of disparate pieces into a new, more powerful way of using networks. However, only after the original ASCII-only browser was replaced by one based on X (Mosaic from the National Center for Supercomputer Applications at the University of Illinois) did the Web really take off. Though originally conceived to integrate existing retrieval and access mechanisms — in particular, the file transfer protocol (ftp), gopher as a menu-oriented retrieval system, and telnet for remote login and interaction with databases — the core WWW protocol (HTTP) has far surpassed usage of all three of these. There are other reasons for the rapid proliferation of WWW, making its rise, in hindsight, a bit less surprising than R. Lucky describes [2]: The technology works reasonably well for access speeds from 2400 baud modems on up since retrievals can be restricted to text only, and newer Internet users, accustomed to graphical user interfaces, are far less tolerant of the command-based interfaces of traditional retrieval mechanisms like ftp or telnet. Also, in the early '90s, the lowest common denominator computing platform shifted from terminals and DOS-based PCs to PCs running windowing systems, X-terminals, and workstations, allowing rapid uptake of Web-based multimedia content, while the basic functionality remained accessible to those still restricted to ASCII. Also, the Internet itself did not have to offer any new capabilities or "service models" beyond a reliable domain name system. Since WWW retrievals are Transmission Control Protocol (TCP)-based, they share the available bandwidth reasonably fairly, and require no new resource allocation mechanisms in the network. Finally, the cost of entry for "consumers" and "providers" alike was extremely low, because the software was (and is) largely free and Web usage, for corporate and university users, incurs no additional network charges. Only now is the sheer quantity of WWW transfers one of the major forces in the increasing congestion in many areas of the Internet, particularly in Europe and the trans-Atlantic connections.

This article will try to present a survey of some of the open areas within the WWW framework, both those that are the subject of current ongoing standardization efforts and those that may impose longer-term fundamental limitations on the WWW. We will investigate the three principal components: the primary transfer protocol (HTTP) in the following section, HTML as the germane Web data type in the third section, and the URL as its naming and addressing mechanism after that. Some ideas on how browsers might develop are presented, and some background is provided on the impact of WWW on the Internet and how it can be made to scale. The section after that points out some longer-term limitations of the WWW model and how other applications could be integrated with the Web. The final section summarizes some of the new applications and alternatives for information delivery that might be visible in the near term. This article does not discuss the important topic of how to organize the resources that can be accessed via the Web; Lynch offers a survey of these more generic issues [3].

[1] See http://www.w3.org/hypertext/WWW/History.html for a timeline of WWW developments.

HTTP



■ Figure 1. *Simple HTTP version 1.0 GET request.*

The "native" WWW protocol for data retrieval is HTTP [4]. HTTP is an application level protocol that is currently used probably exclusively with the Transmission Control Protocol (TCP), although there is no reason it could not be used with other reliable transport protocols. HTTP is a client/server protocol. The client, typically a WWW browser, asks the WWW server for some information via a GET request or, less frequently, transfers information to the server. Currently, version 1.0 (and rarely, version 0.9) of the protocol is in use, with version 1.1 being worked on within the Internet Engineering Task Force (IETF).

A typical HTTP request is shown in Fig. 1. It consists of the request line and a number of parameter-value header lines describing the request (e.g., what kind of data types are acceptable as answers). Each request is handled by its own TCP connection and is completely independent of any previous request; that is, for each document and graphic on a page, the Web browser opens a new connection to the Web server. The server closes the connection to signal that the data has been transferred completely. Given this description, you can build your own Web browser by using a telnet client to connect to port 80 (the standard WWW port) on any Web server and typing the first GET line in Fig. 1 followed by a blank line. The information exchanged by HTTP can be any data type and is not limited to HTML.

This simple protocol has the advantage that clients and servers can be stateless, that is, they do not have to remember anything beyond the transfer of a single document. It is also pseudo anonymous, in that the server sees only the host Internet Protocol (IP) address of the client. Two requests from the same IP address are likely to come from the same individual but, with firewalls and multi user systems, certainly do not have to. For retrieving documents, these two properties are sufficient and desirable. However, they make it difficult to, say, maintain a "virtual shopping cart" for a particular visitor, have an advertising agency track how visitors move through their site, or customize pages for specific visitors. There are at least three approaches to adding state:

- One could generate custom links from a home page so that instead of simply pointing to "chapter1.html," a link would point to "17348/chapter1.html," where the latter has the same content but is specific to visitor 17348. While this scheme works without any support from the browser, it also defeats any caching mechanisms, and does not work beyond a single visit to a home page. Generating custom pages also requires somewhat more processing by the server.
- Forms can contain "hidden fields," which are not visible to the user but can carry values identifying a particular visitor.
- Netscape has proposed "HTTP cookies," where the server response for a page contains a parameter-value pair, an expiration date, and a URL range. The client should then store these and return appropriate parameter-value pairs when accessing the given range of URLs.
- HTTP could be extended to maintain a single TCP connection across several requests [5]. While this is desirable for performance reasons (see below), it does not work across several visits separated by a larger time span.

An observant reader will have noticed in Fig. 1 that the GET request did not contain the whole URL, but rather only the file name part, index.html in this case. While this saves a few bytes and slightly simplifies parsing by the server, it causes problems for the popular approach of "virtual hosting," where a single server "fronts" for a number of URLs; for example, a host host.lpp.com may have aliases www.company1.com, www.company2.com, and so on. Unless it acquires several IP addresses, the server on host.lpp.com cannot tell for which of its hosted URLs a request is. HTTP version 1.1 fixes this by making it mandatory to include the host part of the URL in the request header.

One of the greatest current problems with HTTP is its relative inefficiency. For a typical HTML page, the client first retrieves the HTML page itself, then discovers the potentially dozens of images contained within the page, and issues a separate HTTP request for each. (Cunha *et al.* have measured the average HTML page retrieved at 6.4 kbytes, with roughly seven images per page [6].) Each HTTP retrieval requires at least one round-trip time, plus the TCP connection setup of three one-way delays and a connection tear-down of another two round-trips. Since there is some overlap possible, the smallest retrieval time is four round-trip delays. Thus, for typical long-distance round-trip times of around 300 ms, the minimum latency even with an infinitely fast link would be 1.2 s. In addition, the TCP congestion control mechanism reduces the achievable throughput until the window has been fully opened, again without regard to the link speed [7]. Experiments [8] show that using a TCP connection to transfer only 2 kbytes for a link with 70 ms of round-trip delay results in a throughput less than 10 percent of the best-case value, increasing to only 50 percent with 20 kbyte transfers.

Typical browsers open several TCP connections at once. While this avoids some of the latency induced by the numerous figures on a page, it can also easily overload slow links, since TCP congestion control, based on acknowledgments, cannot control the emission of the first data packet.[2] Thus, most browsers seem to restrict parallel retrievals to about four, also to avoid unfairness (since a user can increase her overall throughput by opening several connections in parallel).

If an item on a page has access protection, an additional exchange is necessary: the server refuses the first request, telling the client how to authenticate itself, and then the client tries again with the proper credentials.

To avoid data corruption due to sequence number reuse, the operating system of TCP endpoints must maintain TCP state information for a few minutes after the connection has been closed. For a busy server, this can add up to thousands of connection records. Some of the TCP-related latencies and the connection record problem can be reduced by a modified version of TCP, called transactional TCP (T/TCP) [9].

HTTP 1.1 suggests extensions that allow a single TCP connection to stay open for several HTTP transfers. This also allows servers to proactively transfer an HTML page, say, with all its icons, without the client explicitly asking for each.

Browsers also reduce perceived latency by showing text before all the images have been retrieved. This, however, is only possible if the image contains hints about the size to be used. This feature is an HTML 3 extension.

HTTP is a textual protocol; that is, all headers are transferred as (mostly ASCII) text. This simplifies the writing of simple browsers, but also increases parsing costs for high-speed servers since the server has to look at every single character to pick apart the header and might have to do some string processing such as line continuations, escaping of special characters, and date parsing. The textual representation for HTTP is also fairly verbose, so the headers can easily be larger than the actual content transferred. As with all Internet

[2] *In particular, the heavily loaded trans-Atlantic links seem to suffer under this high number of short connections.*

Version	Typical browser	Features
1.0	[illegible]	[illegible]
2.0	Mosaic [illegible]	Inline images [illegible]
[illegible]	Netscape, Microsoft	Alignment, tables [illegible]
3.0	Arena [illegible]	Math, floating figures, style sheets

■ Table 1. *History of HTML*

textual protocols, the protocol gets rather more complicated once more than one character set is to be supported. For HTTP, this affects mostly URLs (see the fourth section) and a few less important fields.

The most frequent HTTP operations are GET, PUT, and HEAD (to get header information only). HTTP 1.1 also defines the operations missing to have HTTP take over the remaining functions of the file transfer protocol (ftp): the ability to delete, link, and rename files. Since HTTP offers automatic data compression, format, language, and character set negotiation, and avoids the need to maintain two TCP connections, it seems likely that ftp will gradually be replaced by HTTP. (Currently, there is no explicit HTTP directory command; rather, the client has to rely on the server to generate an HTML representation of a directory listing when the URL points to a directory rather than a file. It would be preferable to have a structured directory representation so the client could decide on the amount of detail, date format, or content representation to be rendered.)

Probably the most notable feature of HTTP separating it from, say, ftp, is its ability to negotiate media types, and (in version 1.1) even character sets and language. The negotiation can be preemptive, where the client indicates preferences for different media types, or reactive, where the server returns a list of possible types. Note that media type negotiation implies that a URL does not necessarily refer to one document, but a single URL could point to the English, German, and French versions, for example. It also avoids having to tailor a server's response to a particular browser make, model, and operating system, so a browser would automatically get the proprietary compression format, while others can obtain a lower-quality "standard" rendition. While these capabilities are rather convenient in many circumstances, it also complicates caching. In an interesting omission, price is not a limit that can be imposed, only size.

Content negotiation can be rather complex, taking into account five different quality scales, and seems unlikely to be amenable to a readily comprehensible user interface. A basic problem is that the client does not know which types the server considers exchangeable for the same URL. If, for example, the user assigns a higher preference to Motion Picture Experts Group (MPEG) audio over telephone-quality audio, and these desirability values happen to be higher than that for text, a clever server may interpret this as saying that the text should be read (which could actually be quite sensible for providing access to a blind audience). There is also an efficiency problem in that the client has to send its complete preference description for every retrieval, since it cannot know which are completely irrelevant. Needless to say, this has not been implemented by any browser or server of which the author is aware. (For servers, it breaks the convenient notion of mapping URLs more or less one-to-one to file names.) For some media types, notably audio and video, additional parameters such as supported sampling rates or pixel depths would be desirable [10], although it seems likely that most systems capable of displaying multimedia objects will soon have the maximum useful capability of 16-bit audio and 24-bit pixels.

There are some efforts to replace HTTP with a binary, ASN.1-based version that supports pipelining of several objects and synchronous retrievals [10]. Since T/TCP and the extensions of HTTP discussed will probably reach a large fraction of the throughput of a revised protocol and the textual parsing overhead is only relevant for the relatively small number of servers, displacement of HTTP by a different protocol does not seem imminent.

HTML

HTML [11] is the "lingua franca" of the Web — the one media type all browsers understand. HTML is a simple document type of the Standardized Generalized Markup Language (SGML). HTML is easy to understand and can be generated by translators from other text formats as well as written by hand. It is bandwidth-efficient and can be rendered in readable form on devices from teletypes and ASCII terminals to high-resolution workstations. Since it contains the actual text rather than font glyphs, it can be translated to, say, braille or synthetic speech.

HTML is a mixture of a presentational and descriptive markup system [12]. Presentational markup systems indicate *how* some text is to be rendered (e.g., in bold face of a certain font, with a given paragraph width). Descriptive markup tags the structure of a document (i.e., whether a certain piece of text is a heading, the abstract, a quotation, etc.). Descriptive markup allows the browser to render content according to the capabilities of the end system such as its screen resolution or the preferences of the reader in terms of fonts, line width, line spacing, and the like. Thus, descriptive markup is inherently suitable for a heterogeneous environment like the Web, where braille readers, ASCII terminals, and small-screen personal digital assistants (PDAs) coexist with 19-inch workstation screens. However, descriptive markup seems to work only for a limited set of documents and has been most successful in relatively technical fields (e.g., for the coding of scientific articles or technical standards). Most Web pages use the HTML descriptive elements to achieve layout and font effects; for example, the lower-ranked headings are used to produce small print.

LaTeX [13] and some troff macro sets also have both presentational and descriptive markup properties, but they differ from HTML in that they are programmable, that is, contain the ability to write (small) programs, declare and invoke macros or functions, or conditionally execute certain parts of the description. This adds flexibility and significantly eases the global manipulation of text. For example, it is possible to define a function that takes a name, e-mail address, and other information as arguments, and then define that function once to render it in different ways (say, as rows in a table or as a list). It is also easy, for example, to render the same text as a single column or several columns without changing the text itself. HTML does not offer this capability. For this and other reasons, HTML is often generated automatically from more capable systems. While this is reasonably straightforward for standard sectioned text, it is rather difficult as soon as tables, equations, or columnar output are desired.

HTML intentionally does not contain these "programming" capabilities since they greatly complicate parsing at the receiver. They would make it very difficult to simply ignore tags a client does not (yet) know. Once they reach the client, HTML documents are also self-contained; that is, they do not reference any external definitions, and thus avoid the problems of missing or incompatible external references. (It should be noted that some servers can dynamically piece together an HTML document with so-called server-side includes.)

Unfortunately, the descriptive capabilities of HTML are limited mostly to low level constructs such as emphasis or indented lists. There are no standardized mechanisms to indicate common document parts such as authors, abstract, keywords, table of content, or references. (HTML 3.0 provides figures with captions.) Abstract, author, and keyword elements would also significantly improve automatic indexing and searching. In addition to the "TITLE" element, it would be very useful to have an element indicating the home (front) page for a particular document, so indices could point to that rather than some random location in the text. In fairness it should be noted that a generic, content oriented encoding even of English text [14] is rather complicated. A particularly relevant omission, however, is document navigation. Almost all HTML pages contain navigation features, that is, links to a "previous" or "next" document, the home page, an index, and the like. However, despite the similar functionality, each document uses its own sets of buttons and links, requiring relearning the "user interface" for each document. HTML elements for navigation would allow browsers to present a uniform, user-configurable interface.

There is work on building full-fledged SGML browsers that could be customized with a document type definition to render any SGML document.

HTML is oriented towards display rather than printing or storage. First, HTML follows a "scroll model"; it has no notion of pages, which is appropriate for display but makes it difficult to print pleasing output. Also, the appropriate delivery units for display are small to reduce transfer latency and scaling, but it means that a single article has to be pieced together from numerous HTML files to be printed or saved locally, each piece likely containing rather distracting navigation icons. The print quality is further diminished by the lack of a vector-based graphics standard within the Web; the common graphics formats Graphical Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and X bitmaps all render bitmaps and usually look rather bad on printed output where the resolution is three to ten times higher. Chaining of page sequences and standardized HTML navigation tags would avoid the necessity of maintaining a separate viewing and printing representation of the same text.

Because of its nonprocedural nature, HTML parsers can be rather forgiving of any HTML coding violations. A large fraction of Web pages, even those professionally authored, contain coding "mistakes," often used to achieve particular layout effects on popular browsers.

Only since November 1995 has there been an IETF standard for HTML [11]. However, a large fraction of HTML pages use various supersets of this standard for such things as to display tables, colored and textured page backgrounds, or fonts of different sizes. Many commercial servers have started to interpret the HTTP header field identifying the browser software release to deliver custom-tailored renditions of their material.[1] Clearly, this does not scale as the number of browsers increases.

Each new browser and browser release seems to introduce a new set of HTML tags or new parameters to older tags. While browsers usually just skip unrecognized tags, a conscientious content developer still has to test the material with a bevy of different browsers to make sure it looks acceptable on all. Most of the new tags seem geared to satisfying advertisers.

This has yielded the blink tag, marquees that stay on the screen even when the text is scrolled, and audio that plays automatically (and continuously) whenever a particular page is viewed. There seems to be little urgency in getting tags for mathematical typesetting into browsers, but it has been defined as part of HTML 3.0.

In both the descriptive and presentational camps, there are efforts to integrate other text formats into the WWW environment. This usually just requires a way to embed links in that text format and somehow integrate the display engine with a browser. There is also work on building full-fledged SGML browsers that could be customized with a document type definition to render any SGML document, including those in HTML. However, the syntax of SGML was clearly not designed by compiler writers; it is not representable as a regular expression or in other simple languages. SGML allows a number of abbreviations and tag omissions whose full support is rather cumbersome. A document type definition requires arcane expertise and is unreadable (and probably semantically incomplete) without copious comments. It also has some strange remnants concerning spaces, character sets, and line endings dating back to punched cards and fixed-length records. Despite its shortcomings, it is unlikely it will be replaced any time soon.

A somewhat more modest (and complementary) effort adds so-called style sheets to HTML. Style sheets describe how a particular tag is to be rendered (e.g., that a heading is to have a certain color or font size). This relieves some of the pressure on HTML to provide new tags for new presentation forms, yet allows page designers and readers control over the rendering of material. Currently, many high-end graphical pages use bitmaps for full layout control. Bitmapped pages and additional text types have one severe disadvantage: they cannot easily be automatically indexed by various "web spiders" and search engines. Also, it is not easy to have a single client-side search if there are several different text formats.

HTML basically has three kinds of hyperlinks. The first wraps a block of text or an image in an anchor HTML tag. Typically shown underlined, it leads to the replacement of the current Web page with the one to which the link points. Since there is only one display, it imposes a linear navigation sequence, where it is difficult to retrace one's steps beyond a few mouse clicks. (History lists are also generally only linear, so side excursions quickly make one lose one's place in the web of links.) The second, not yet standardized, displays the document pointed to in a new browser window. Finally, the IMG tag, not generally perceived as a link, displays some image in-line as part of the document. Only recently have there been efforts to generalize the in-line display so that almost any data type, including active content like Java applets or animations, can be part of a document.

Also needed are "lightweight" anchors invoked simply by leaving the pointer on a page position and displaying a small, transient window with a small amount of information. These would be appropriate for displaying, say, a brief help message for a button or the definition of a word, similar to the "balloon help" feature in some operating systems. Similarly, it would be helpful to be able to define a default link so that a region of text is passed to a search engine defined by the reader or the page creator (e.g., a dictionary or a translation engine).

[1] *This has even led a large PC software vendor to identify its product to the server as that of its competition.*

Currently, links can only be added by the author of a document. A different model of hypertext separates document and links, so a single document can feature different sets of links, and links can be updated without touching the document itself. This also allows annotations to be added by readers. Annotations were an original experimental feature of WWW that seem to have disappeared almost completely. Certainly, anything but personal annotations raises all kinds of liability issues. Because the content of the same URL often changes or a single URL can refer to several documents, as discussed earlier in the context of content negotiation, it is also difficult to come up with a reliable, persistent annotation mechanism.

> Within the Internet, HTML is already replacing a number of similar text systems such as MIME rich text or troff manual pages.

Within the Internet, HTML is already replacing a number of similar text systems, such as multipurpose Internet mail extensions (MIME) rich text [15] or troff manual pages.

While there have been extensions of SGML to the presentation of continuous media (HyTime), they appear overly complex but still do not offer the full programming flexibility of a client side programming and scripting language such as Safe-Tcl or Java. Proprietary solutions, particularly those that have widely deployed authoring tools, seem to be dominating this application.

URLs and URNs

Universal resource locator is just one of the names used to designate objects within the World Wide Web. The whole family is technically known as universal resource identifiers, of which URLs name the physical location of an object [16, 17], universal resource names (URNs) the identity without regard to location, and uniform resource citations (URCs) describe properties of the object. Only URLs are in widespread use. They consist of an identifier for the protocol (http, ftp, etc.), the network location (host and port), and a path name within the named server. While that path is typically mapped directly onto a file name by the server, this is purely a server convenience. A server could just as well use this path as a key into a database or as a function name and arguments to dynamically generate a document.

Note that a URL says nothing about the type of object to which it points, even though most URLs give some informal hints to the initiated. An example of a URL is `http://www.w3.org/`. Most home-page URLs pass the "business card test," the failure of which doomed X.400-style email addresses; but they are still punctuation-heavy for reading over the telephone or radio. This has led to the http being dropped as the default and browsers assuming that any URL starting with www is to be accessed via HTTP.

Despite its apparent hierarchical nature, the domain name space (DNS) is practically flat because the large majority of names are drawn from the .com domain. Even countries as large as Germany have a flat second-level domain name. Since registration costs only about $50 a year, companies have taken to registering every one of their products as a domain name. This is relatively harmless, but causes inevitable clashes since the Internet domain name space lacks the distinction of type of trade, country, and region that allows reuse of popular names in the area of traditional trademarks. Thus, the regional naming structure found in the .us domain will have to become the common case rather than the exception.

Letting a URL name a single host has a number of disadvantages. If that host crashes or is overloaded, the retrieval fails. Large sites have found a number of work-arounds like DNS aliases or distributing different name information to different subsidiary name servers.

In a better system, the domain name service used to translate Internet host names into IP network addresses would offer a per-domain service listing, extending that used for e-mail today. Most organizations have a so-called mail exchange record which names one of most hosts responsible for all e-mail for that domain. A mail transport agent goes down the sorted list until it finds a working remote host. A similar system could be introduced for WWW service. There are also currently proposals to include a host's geographical location in its DNS record. Given a list of alternate, equivalent servers and their geographical location, browsers could look locally first before sending a request across the oceans.[4]

URLs were often considered to be temporary artifacts until a more powerful naming mechanism could be deployed. However, URLs seem to be experiencing the same longevity as e-mail addresses, also long condemned to replacement with lookups by name in a global X.500 directory. The basic idea has permanent names, URNs, lead through a directory to URCs describing their properties and URLs listing their current location. However, somebody has to maintain and pay for this directory. The same URLs that change frequently are also likely to be the ones that the maintainer forgets to update in the directory since she has lost access to or interest in the content. Finding documents would only be made easier if the number of registries is relatively small and if authors of documents, including all the mixed-media documents, VRML worlds, and the like, can agree on a common set of meta data (URCs). For electronic publications of well-known publishers and scientific societies, the addresses of individual documents within their libraries are not very likely to change. Thus, it appears unlikely that users will be willing to pay, in either monetary units, additional lookup delay, or decreased reliability, to avoid "stale" links. For some forms of documents, particularly university technical reports and scientific periodicals, an electronic directory seems appropriate and likely to take shape, since the organizational infrastructure and incentive are largely in place. Also, it would indeed be very useful to have a "repository of last resort" for digital artifacts, the equivalent of the U.S. Library of Congress.

Browsers

One of the factors driving the success of the WWW is its ability to both attract content providers and serve as a base for new applications. While in the past a corporate library may have written its own user interface to its library catalog, it now appears much easier to base this on a Web server and browsers. This avoids having to write a new user interface for each new client platform or operating system, and automatically lets the system participate in advances like security.

There seem to be two contradictory directions for WWW applications: the browser that can do everything and having every application have WWW capabilities. The latter makes it difficult to integrate several data types, but it is certainly desirable to have applications recognize URLs and control a browser to view the URL content. Browsers are already incorporating mail tools, news readers, and very primitive file system managers. Soon they will also feature text editors, at least for HTML. While this integration has the advantage of hiding the difference between local and remote operations to a large extent, it also leads to huge applications and less choice of vendor. Other mechanisms to integrate different applications are currently being deployed, for example, so-called plug-ins, where applications communicate directly with the browser and share some of its window area.

> There seem to be two contradictory directions for WWW applications: the browser that can do everything and having every application have WWW capabilities.

Network Impact and Requirements

Since the Internet has gone commercial, it is difficult to ascertain exactly what fraction of wide-area traffic is generated by WWW browsers and HTTP; but informal indications are that the fraction is well above half the total traffic. WWW stresses the Internet in that browsing has low latency requirements, as a human is waiting impatiently. The data transfers can be anything from a short burst for a small image to several tens of megabytes for a video or audio clip. While text and images on a single page usually come from the same host, hyperlinks traversed have no real spatial correlation. The low spatial correlation will strain any IP-over-asynchronous transfer mode (ATM) mechanism that attempts to set up individual switched virtual circuits for each Web retrieval.

For any of the more popular or high bandwidth services, the WWW can only scale if information content is mirrored and cached. A mirror provides a complete copy of some server, with the master server explicitly updating its mirrored copies. Mirrors are trusted, at least to some extent, by the data source. Mirrors can be configured by manual selection or, possibly, through a domain name service mechanism. If URN directories ever come into widespread use, these could obviously also return several candidate locations for a document. A busy server could also send a redirect answer to a browser, but it had better be sure that the server to which it is redirecting users is not busy or down.

Caches are placed between client and WWW server (e.g., on the premises of a company or a university) and have no direct trust relationship with the server. Typically, the browser is manually configured with the location of a single cache server. Caches may connect to other caches. The client connects to the cache and requests the document, which the cache server either has stored already or in turn retrieves from the actual server, keeping a copy to satisfy future queries. Caches can be quite effective [6], but can also easily become bottlenecks and then be avoided by users. Also, many documents that look static are actually generated anew for each request, and thus are not cacheable. Since many servers want to keep a running hit count for their offerings, they actively defeat caching by setting expiration dates as immediate or otherwise marking information as noncacheable. Thus, some mechanism must be found to enable caches while providing accurate access counts to the main server. Also, documents with access authorization currently cannot be cached; in the future, this may be a large fraction of popular documents.

If a hierarchy of caches is to be built, there has to be a routing mechanism that determines which cache(s) should be queried for a document. For that, a client may need to know the actual location of the document so it can avoid contacting a cache where the actual location of the resource is closer. It remains to be proven how much can be gained with multiple cache levels since there is probably a stronger affinity of interest on the level of a company or department than a whole country.

Limitations of the WWW Model

Despite all the press and publicity, the WWW model is currently rather limited: retrieve an object (text, audio, or video) and render it. Even with forms, the capabilities of a Web page are roughly that of a page-oriented mainframe terminal, with some graphics spice added. Some inherent capabilities of the Web model have not been developed, in particular the ability to store content through the server. This could be quite useful for collaboration and for maintaining corporate information within intranets[5] and across firewalls, particularly once client authentication is better developed. It is likely that future browsers will cease to be display-only and allow editing and storing back documents, at least those written in plain ASCII or HTML. This would make them more competitive with other asynchronous computer-supported cooperative work environments.

Client-side interaction is currently limited to filling out simple forms and clicking on buttons and bitmaps (so-called image maps). There are some efforts to provide more direct feedback to the user than having to fill out the whole form only to be told that some field is wrong or clicking on parts of a bitmap without any feedback as to what, if anything, might happen. Client-side image maps store the coordinates of sensitive areas so the browser can provide local feedback. Client-side scripts or applets would allow the provision of interactive help or correctness checks, as well as possibly richer user interfaces so the content of a page can adapt to user action rather than simply reloading a page from the server.

The integration of multimedia is currently very primitive. A video or audio file is transferred via HTTP and then played with sufficient buffering or from local temporary storage. Playing audio and video as it arrives from the network avoids waiting minutes for it to download completely (only to discover it probably was not worth the wait). However, unless the minimum network bandwidth is the access bandwidth, assumed known, the user has no way to choose an appropriate encoding or know ahead of time for how long the media content needs to be buffered to ensure playout without interruptions. A number of solutions can be envisioned. First, non-TCP protocols such as Real-Time Transfer Protocol (RTP) [18] that provide congestion feedback can be used to adaptively tune their buffering and encoding to the available bit rate. In addition, for both TCP and other protocols, resource reservation could guarantee a minimum bandwidth [19]. Adaptive applications require no changes to the Internet, but are still subject to glitches when adapting quality down-

---

[4] This assumes that geographical distance is equal to network distance. However, the shortest network distance between two European cities is often through Washington, D.C.

[5] Intranets are networks within organizations that may or may not be connected to the Internet, but use some of the same technology and protocols.

ward. RSVP, the IETF Internet resource reservation protocol [19, 20], will likely not be deployed widely for a number of years, because it needs router modifications, some form of authentication, and usage-based billing to prevent abuse.

Any type of interactive games (often called "twitch games") will likely bypass the Web protocols completely, even though they might use one to locate services or partners to play with. The same is also true for multimedia on-demand applications. The existing media-on-demand applications use the WWW browser only to present some VCR-style controls and a play list. There is no possibility of embedding "active areas" (i.e., hyperlinks) within the audio or video content or synchronizing textual and video content.

For content delivery, the Web is "pull-only"; that is, the browser has to explicitly check whether there is new content. There are some HTML extensions where the browser can be instructed to do this at preset intervals, and browsers can be told to check a subset of bookmarks for any changes, but neither is particularly efficient. There are also services that regularly poll Web pages for changes and then send e-mail to the subscribers of this page, saving unnecessary queries for popular servers. As an alternative, it would be close to optimal in terms of bandwidth usage and speed of notification to have servers send announcements by IP multicast to all interested parties. This, however, works only for systems continuously connected to the Internet.

Web browsing is a solitary occupation. A visitor to a Web page cannot see if there are other like-minded people reading the same page or interacting with the same 3D world. There have been a number of proposals to create mutual awareness. The Virtual Places [21] protocol uses a separate server to track visitors, represented by small cartoons or pictures displayed on a custom browser. A visitor sees the movement of others on the page and can move her image to touch that of another visitor. As images touch, a packet voice connection is established. Tour buses can be used to visit Web sites together, with "passengers" on the bus reachable by audio and video through IP multicast. People who are on the same page can also "chat" with each other by typing messages. However, all of these interactions only use the browser as a platform and bypass WWW protocols. Such Web-based interactions might also be feasible by using scripting languages and multicast notifications of presence and location. This offers an enhanced version of the room metaphor employed in some video conferencing systems [22].

Chat rooms are quite popular on on-line services, and there have been attempts to use WWW technology. Without some external protocols this is rather clumsy, as a user would have to reload a page to see what, if anything, others have typed.

New Applications and the Competition

The simplicity of Web technology is in sharp contrast with some other efforts offering networked multimedia, such as those building on the MHEG description, an International Standards Organization (ISO) standard [23, 24]. MHEG tries to provide a generic multimedia model that supports data types from text to synchronized audio and video as well as dynamic user interaction elements. However, producing even simple information requires specialized authoring tools. To offer more than text, much higher access bandwidth is needed, thus limiting this to video-on-demand applications running over cable TV and set-top boxes. For many of these, it would seem sufficient to simply send back timing and pointer coordinate information to the server rather than having the set-top box interpret MHEG "programs."

In the early 1990s, there seemed to be a rush by every cable and local telephone company to trial various forms of video-on-demand services. Some of the additional services, such as home shopping and banking, would arguably be similar to what is now slowly emerging on the Web, but the pool of providers would have been strictly controlled by the cable or telephone operator. Experiments showed that most households only watched three or four paid movies per month. Given that the premium over walking to the neighborhood video rental store has to be fairly small (unless there is a blizzard outside), the additional revenue turned out to be rather small, not enough to compensate for the large investment in video servers, set-top boxes, and network infrastructure. It remains to be seen whether cable modems, allowing packet traffic from cable headend to subscriber at around 10 Mb/s shared between all subscribers on that cable segment and a much lower upstream data rate, are a viable alternative. They would seem to offer the bandwidth for small-scale video-on-demand and near-video-on-demand services via WWW protocols as well as immediate access to the Internet at large, while avoiding the customer having to purchase a special-purpose computer disguised as a set-top box. For these cable modems, very cheap "Internet appliances" or slightly enhanced video game machines may also be attractive for the large majority of households that do not own a personal computer.

However, should video on demand be deployed, the digital infrastructure would also be usable for Internet services. The use of WWW with a set-top box and a TV as a monitor is possible, but unlikely to find widespread use. Despite increased resolution of recent models, TVs still cannot reproduce a 640 x 480 image, the minimum acceptable for displaying WWW pages with significant amounts of text. Clicking on hypertext links could be difficult because they are often too close together to aim at with an air mouse; however, an interface similar to ASCII WWW browsers, using arrow keys to move from one hyperlink to the next, might be workable. Textual input would be extremely tedious unless the remote control featured a small keypad — a good role for a personal organizer with an infrared interface.

The roles of on-line services relative to the Web has seen much discussion, with some dismissing the on-line service providers as parts of a bygone area. Certainly, a number of new services seem to be heading towards being "gated Internet communities" rather than using proprietary technology, shifting their focus from providing content to providing Internet access. However, the large majority of residential users still use on-line services, with occasional forays into the Internet. On-line services not only provide access, but also:

- Technical customer support services
- A basic menu of standard content
- Aggregated, averaged billing for a range of content
- Parental access control features

In particular, the value of providing billing services should not be underestimated for both content providers and consumers. A single, predictable bill for all content makes a service more attractive, even if an individual actually uses only a fairly constant tiny fraction of it. Besides advertising support, there does not seem to be a viable means of creating revenue for individual Web sites. Most people currently do not spend enough time at a single site to make subscriptions attractive. (An average customer of an on-line service spends about six to eight hours a month on line.) Pay-per-view services have generally not been very successful in other areas, unless content could not be obtained elsewhere. Efforts to develop micropayment schemes might provide a way to charge some small amount for each page retrieved, but nobody knows how users will react to "hearing the meter ticking." The September Project is investigating some of these issues in the context of scientific publications [25].

On corporate intranets, Lotus Notes™ has been used for many of the same functions as the Web. While the basic access mechanisms are similar in spirit, if very different in detail, to a combination of e-mail, Usenet newsgroups, and HTTP document retrieval, the backend differs. While most Web servers base their document structure on the local file system, Notes uses a distributed database system containing compound documents. The individual databases are synchronized periodically or on demand on a per-record basis. Similar to the just developing server and client side scripting capabilities of the Web such as cgi-bin scripts, Java, and Javascript, Notes has various languages and application programming interfaces to construct business applications, and tools to develop them. The Web server script industry is still very much in its infancy. Notes servers have the capability to integrate with corporate mainframe databases, something Web servers are only just developing. The Web still needs to develop the workflow capabilities offered by Notes. (Workflow capabilities automate the typical flow of forms and correspondence for anything from purchase orders to personnel record changes and travel accounting.) Unfortunately, it is likely that this capability will be just as proprietary in the Internet as in the Notes environment. Thus, for publishing relatively static documents and simple scripts, WWW and Notes offer similar capabilities, while Notes seems to have a headstart on the "backend" (i.e., server) side. Recently, Notes databases have become accessible from Web browsers [26].

Conclusion

The success of the World Wide Web has, at least in the eyes of the nontechnical public, made it and the Internet nearly synonymous. However, there are many interesting Internet-based services that are largely independent of the Web and have quite different requirements. As an example, real-time interactive multimedia services would provide an alternative to the telephone network, immediately offering many of the services promised by the telephone carriers through advanced intelligent networks (AINs), but they require extensive additions to network infrastructure, particularly some form of resource reservation and significantly enhanced bandwidth. A Web-centric Internet would be giving up one of the fundamental strengths of the Internet: the ability to quickly deploy new services.

Overall, the continued growth and success of the WWW as a global delivery mechanism for multimedia content will ride on technical issues as much as on economic, social, and political ones. Some questions that we can only raise here include: Can an advertising-only medium prosper, or will there be ways (and willingness) to pay for quality content? Will jurisdictions force Internet service providers to restrict access to parts of the Web considered in violation of local criminal statutes? Will encryption and user certification be widespread and sufficiently easy to use that electronic commerce can prosper?

References

[1] T. Berners-Lee et al., "The World Wide Web," Commun. ACM, vol. 37, Aug. 1994, pp. 76–82.
[2] R. W. Lucky, "Where Did the Web Come From?" IEEE Spectrum, vol. 32, July 1995, p. 15.
[3] C. A. Lynch, "Networked Information Resource Discovery: An Overview of Current Issues," IEEE JSAC, vol. 13, Oct. 1995, pp. 1505–22.
[4] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol — http/1.0," Internet draft, IETF, Aug. 1995, work in progress.
[5] R. Fielding, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol — http/1.1," Internet draft, IETF, Nov. 1995, work in progress.
[6] C. Cunha, A. Bestavros, and M. Crovella, "Characteristics of WWW Client-Based Traces," Tech. Rep. BU CS 95 010, Boston Univ., Comp. Sci. Dept., July 1995.
[7] W. R. Stevens, TCP/IP Illustrated: The Implementation, vol. 2, (Reading, MA: Addison-Wesley, 1994).
[8] V. N. Padmanabhan and J. C. Mogul, "Improving HTTP Latency," Proc. 2nd World Wide Web Conf. '94: Mosaic and the Web, Chicago, IL, Oct. 1994.
[9] R. Braden, "T/TCP — TCP Extensions for Transactions — Functional Specification," RFC 1644, IETF, July 1994.
[10] S. Spero, "HTTP-NG Architectural Overview," electronic doc., Feb. 1995.
[11] T. Berners-Lee and D. Connolly, "Hypertext Markup Language 2.0," RFC 1866, IETF, Nov. 1995.
[12] J. H. Coombs, A. H. Renear, and S. J. DeRose, "Markup Systems and the Future of Scholarly Text Processing," Commun. ACM, vol. 30, Nov. 1987, pp. 933–47.
[13] L. Lamport, LaTeX: A Document Preparation System, (Reading, MA: Addison-Wesley, 1986).
[14] C. M. Sperberg-McQueen and L. Burnard, eds., Guidelines for Electronic Text Encoding and Interchange, (Chicago and Oxford: Association for Computers and the Humanities, May 1994).
[15] N. Borenstein, "The Text/Enriched MIME Content-Type," RFC 1563, IETF, Jan. 1994.
[16] T. Berners-Lee, L. Masinter, and M. McCahill, "Uniform Resource Locators (URL)," RFC 1738, IETF, Dec. 1994.
[17] R. Fielding, "Relative Uniform Resource Locators," RFC 1808, IETF, June 1995.
[18] H. Schulzrinne et al., "RTP: A Transport Protocol for Realtime Applications," RFC 1885, IETF, Nov. 1995.
[19] L. Zhang et al., "RSVP: A New Resource ReSerVation Protocol," IEEE Network, vol. 7, Sept. 1993, pp. 8–18.
[20] R. Braden, L. Zhang, and S. Berson, "Resource Reservation Protocol (RSVP) — Version 1 Functional Specification," Internet draft, IETF, Nov. 1995, work in progress.
[21] E. Shapiro, "Virtual Places — A Foundation for Human Interaction," Proc. 2nd World Wide Web Conf. '94, Chicago, IL, Oct. 1994.
[22] S. R. Ahuja and J. R. Ensor, "Call and Connection Management: Making Desktop Conferencing Systems a Real Service," ACM Comp. Commun. Rev., vol. 22, Mar. 1992, pp. 10–11.
[23] T. Meyer-Boudnik, "MHEG: ein Austauschformat für interaktive multimedia-Presentationen," Tech. Rep. TR-94-002, Praktische Informatik IV, Universität Mannheim, 1994.
[24] R. Price, "MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange," Proc. ACM Multimedia '93, Anaheim, CA, June 1993, pp. 121–28.
[25] H. Schulzrinne, J. Brassil, and A. Choudhury, "Internet Access to IEEE Journals: The September Trial," Proc. Documents in Public Networks — Applications and Experience (Dokumente in öffentlichen Netzen—Anwendungen und Erfahrungen), Darmstadt, Germany, Nov. 1995.
[26] M. Zisman, "Lotus Notes Web Strategy," transcript of press briefing, Dec. 1995.

Biography

HENNING SCHULZRINNE received his undergraduate degree in economics and electrical engineering from the Technische Hochschule in Darmstadt, Germany, in 1984, his M.S.E.E. degree as a Fulbright scholar from the Univ. of Cincinnati, Ohio, and his Ph.D. degree from the Univ. of Massachusetts at Amherst in 1987 and 1992, respectively. From 1992 to 1994 he was a member of technical staff at AT&T Bell Laboratories, Murray Hill, New Jersey. In 1994 he joined GMD Fokus (Berlin, Germany) as a postdoctoral researcher. His research interests encompass real-time network services, the Internet, and modeling and performance evaluation.

ATM Networks

AASERT 1996 Summer Research Program
ATM NETWORKING TEAM FINAL REPORT

Dr. Linda Hayden, P.I.
Mr. Darnley Archer, Mentor
Mr. Wayman White, Mentor
Sharon Saunders, Graduate Student
Derrek Burrus
Curtis Felton
Shanita Powell
Antonio Rook

This summer the ATM Networking group discussed some theoretical concepts of ATM and the ATLAS program. The team also focused on other topics such as networking faculty offices, becoming familiar with UNIX commands and file system, and reviewing two articles on current technology taking place throughout the nation.

I. ATM

The concepts of ATM that were discussed were its architectural/ transmission views, its connectivity, and the cell itself. The three architectural/ transmission views compared and discussed were packet switching, frame relay, and cell relay.

Packet switching is a method of transmitting data messages through a communications network, in which large data is broken into smaller packets. Data is transported across a medium in packets. These packets are then transformed into frames, where they are converted to packets. Once reaching their destination, the packets are changed back to frames, then to packets. (See Diagram 1) Packet switching transmits data on a "first come, first serve" basis making the transfer time vary.

• Packet Switching

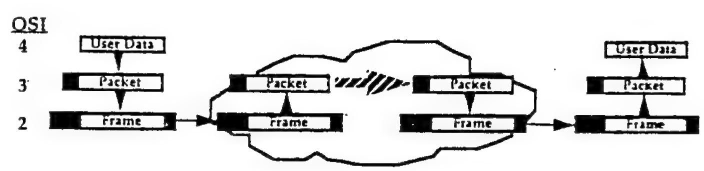
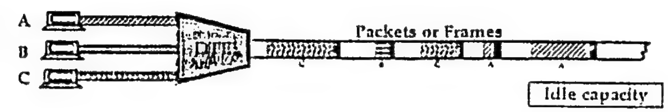


Diagram 1

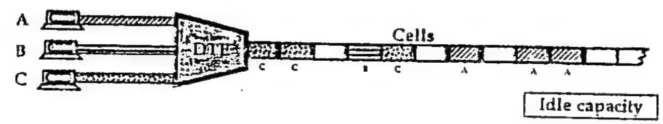
Frame relay is an updated type of communication network from packet switching. Data is transported in frames as oppose to packets and is transported quicker to its destination. When errors are found the frames are discarded and the user must retransmit data. Frame relay is somewhat similar to packet switching because both transmits data on a "first come, first serve" basis and the amount of time it takes to transfer information varies.

• Packet Switching
• & Frame Relay



Cell relay, an improvement of frame relay, is the most commonly used transmission for ATM. Information is broken down into fixed "cells" of 48 bytes that can be easily transported without a high risk of losing data. It also transmits data on a "first come , first serve" basis, but transmission time is quicker because of the fixed length cells. Cell relay has a priority scheme which allows some data to have higher transmission priority. In most cases, video and audio carries a higher transmission priority than data.

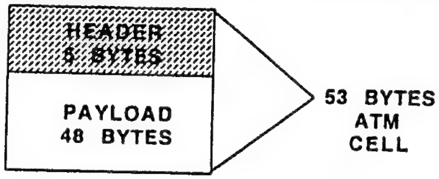
• Cell Relay



The next part of ATM discussed was connectivity. Connectivity is made up of three parts: physical link, virtual path(VP), and virtual channel(VC). The virtual path describes a set of virtual channels that are grouped together between cross points. Virtual channel describes the flow direction of ATM cells between connecting points that share a common identifier number. The VP and VC is the route that the data is transported from point to point.

The ATM cell is 53 bytes long consisting of two major parts, a header and the payload. Each cell has a 5-byte header that identifies the cell's route through the network. It also has 48-byte payload of user information as well as service adaption functions. This user data in turn carries any headers or trailers required by higher level protocols. (See Diagram 2)

Diagram 2



In preparation for bringing ATM and Ethenet to the desktop in Lester Hall, the following was done. The communication closet in Lester Hall was set up as such, 5 hubs were installed given us 120 ports available for Ethernet to the desktop. In order to link the hubs, we had to install two types of EPIM cards, EPIM-T (twisted pair) and EPIM-F2 (fiber optic) into the hubs. We used a cable of 15 pairs of fiber that were pulled from Doles Hall to Lester Hall's communication closet into the Fiber Distribution Center (FDC). A fiber optic patch cable is connecting the very first hub from the FDC. Also, there was a twisted pair patch cable attached from hub to hub to give connectivity. Next, we had to make twisted pair jumpers to go to the patch panel from the hubs.

The FDC distributes the fiber to its destination. From the FDC, the patch cable goes to the ATM switch. The purpose of the switch is to convert data to ATM speed. A patch cable is then connected from the ATM switch to the Ethernet switch, which sends data through Ethernet line versus fiber optic. Finally, the ethernet switch is connected to the rack of hubs already installed. At the present time, data is being sent via ethernet to the desktop.(See Diagram 1 in Appendix A) Future plans to got ATM to the desktop is to add a patch panel in the communication closet and another in the lab. These patch panels will be connected with fiber.

II. ATLAS

ATLAS is an acronym for Affordable Technology to Link America's Schools. The main objective of the ATLAS program is to enhance the economic competitiveness of tomorrow. This project is designed to allow K-12 schools the opportunity to have internet access. There are four key entities in the implementation of the ATLAS program. They are NASA, state governments, national institutions, and industries. NASA center's role will be to obtain state

government buy-in, offer partnership roles to the State Department of Education, commercial sponsors, etc. The state government will address the need for ATLAS to be implemented across the state and also to identify universities, governor schools, and other organizations which could serve as Internet Central Sites. The industry's role is to identify the functions of ATLAS technology and provide a demonstration of how it can be supported and maintained by their company.

The architectural design of ATLAS is to have a server, within the K-12 schools. This server will serve as a internet host for that school. It will have a modem attached that will allow the school to have dial in access to the host site. The server at that host site is then connected to the Internet. K-12 schools get their access via a host site. These connections can be seen in Appendix B. Diagram B-1 shows the Local Area Network (LAN) within the K-12 schools. Diagram B-2 shows the Wide Area Network (WAN) using the host site as the internet provider.

The advantage of ATLAS is its use of caching. The server in the elementary and secondary school has a external harddrive for caching connected to it. An example of cache is the storage of data to be used at a later time. The advantage of the caching system is the control it gives the school over data being broadcasted in and over the school. It allows the students to retrieve information and store it on the external harddrive. This information can later be used by other students which keeps the use of the modem line down to a minimum.

The government funds the ATLAS program, however they only fund the research on an assessment of what a school has and what will be needed to run the ATLAS program at that school. The elementary and secondary schools pays for all the equipment and of the training. NASA and host sites pay for the remainder of the training.

The team visited three K-12 in Portsmouth, Va. (Emily Spong Elementary, Douglas Park Elementary, and I.C. Norcom High School) that are a part of the recently funded grant from NASA. The purpose of the visits were to see how they could ttake advantage of the ATLAS program. The visits consisted of notating and documenting their current electrical outlets, computer types, and other things in their computer labs. The purpose was to inform the schools them on how their labs should be setup. It also included the types of hardware and software needed in order to run certain applications such as Netscape.(Diagrams of each school can be found in Appendix C.)

After an assessment of Emily N. Spong Elementary School's technology, the following conclusions have been made. The library has been selected to serve as their computer resource lab. The lab consists of ten Macintosh LC II's, a 6100/66 Power Macintosh, and an ImageWriter II printer. The Macintosh LC II's currently have two expansion slot cards with one

slot being used for 5.25 external floppy drive. The LC's can be upgraded to meet the standard of ATLAS by adding disk space, RAM, and Ethernet card for networking purposes. A total of four lines is suggested to connect the ATLAS server.

At Douglas Parks, there are a few key factors that were needed to be noted. First, we decided that the phone line in the Library would more than likely be the line connected to the ATLAS server. There are currently 28 macs being considered for the ATLAS program; 22 LCII's, 4 LC 575's, and 2 mac laptops on order. It was recommended that the lab hold at least 15 computers to comply with the average 30 students per class. This makes access to the computers easier by assigning two students per machine. The remaining computers will be distributed throughout the other classrooms, utilizing one as a teacher workstation. There is also the possibility of setting up floating machines on cart to allow portability.

After assessing I.C. Norcom High School's technology, these conclusions have been made. Currently, there are two options as to where the ATLAS server can be placed. It can be put in the library (room 211), located on the 2nd floor, or the computer lab (room 108), located on the first floor.

In the lab there are 15 computers; 1 IIvx, 11 LCII's, 1 Quadra 800, and 2 LC's. All the LCII's have a 440 harddrive, the LC's have a 240 harddrive, and the Quadra 800 and IIvx have 8megs of RAM and a 240 harddrive. Plans are being made to add five more computers to the computer lab.

Once all the assessments were made, a list of proposed items that are required in order for all the mentioned K-12 schools to have Internet access was composed. The list consist of the following:

1. Minimum of 15 Macintosh systems
2. At least 16MB of RAM for each machine
3. Telebit Fast Blazer 28.8 Modem
4. SCSI External Drive (cache, 2.1 GB)
5. Hub and cables
6. Ethernet LAN Networking Card
7. Networking software (Network starter kit (optional))
8. Server, consisting of:
    * Sunsparc 4
    * 535 MB of internal Harddrive
    * 32 MB of RAM
    * Color Monitor
    * Internal CD-ROM Drive
    * Internal Floppy Drive
    * Multiport Magma Serial Card
9. Three phonelines for administrative staff and teacher use in addition with the phoneline to dial out to the server at the host site.

III. Networking Faculty Office

Networking the faculty offices is one of various tasks to be completed for this summer in order to give professors access to the Internet from their offices. In order to set a PC up on the web, we had to install the Network Starter Kit Software. The directions for installing starter kit and netscape will follow:

Directions for running starter kit

1. Run ezstart (if not installed then install using disk)
   *(note the RAM address) to verify the x= line in #2

2. Modify config.sys
   line 2 = c: \dos\emm386.exe noems x=CC00-CFFF
   (may change according to machines address)

3. Edit autoexec.bat
   Add the following lines at the bottom of the file:
   cd \smcpck
   pack1
   cd \
   *(if there is a window or menu in the autoexec.bat file then add the 3 lines before those lines)

4. Create directory called smcpck
   type command: (mkdir smcpck)

5. To Copy information from driver disk to smcpck directory:
   type command:(xcopy *.* c:\ smcpck)

6. Install starter kit
   All instructions in starter kit book start on pg.7 section 1.3.1 then skip to Section 1.3.3

For network starter kit running TCP-MAN

1. Go to "File", Run, TCPMan under Winsock
2. Enter IP address

Netmask: 255.255.255.0
Name Server: 152.4.20.3
Default Gateway: 198.85.48.254
Domain Suffix: ecsu.edu
Packet Vector 7e

3. Exit
4. Go to File, New, Program Group and title it Network Starter Item
5. click on main, then windows setup
6. Options, Setup applications, search for applications, c: local drive
7. Select following files and select them by pressing the spacebar:
   - D shell
   - eudor 144
   - FTP LPQ Utility
   - FTP LPR Utility
   - FTP RSH Utility
   - ftpw.EXE
   - hopchkw.EXE
   - MOSAIC
   - pingw.EXE
   - tcpman.EXE
   - telw.EXE
   - trmptel.EXE
   - view.EXE
   - winarch.EXE
8. Click o.k. continuously until set-up is complete
9. Copy tcpman.exe into the startup folder

Installing Netscape 2.0 (optional)

1. Go to Program Manager and select Main, put disk in
2. Change to a: or b: drive
3. Tile screen under Windows menu
4. Go to root directory and create a directory called netscape
5. Open the directory
6. Copy files from a: or b: drive to the netscape directory by holding the shift-key and use arrow keys to select files
7. Redo no. 6 for disk 2
8. Double click on setup.exe in netscape directory
9. During setup keep clicking next until it stops loading

10. After the setup is completed, return to the Program Manager

Aside from networking, the team is also responsible for system administration tasks and duties therefore, being conscious of commands and file systems is a necessity. The two UNIX books we used were UNIX Tamed by Rodney Wilson and UNIX Systems by Douglas Troy. These books included questions and exercises demonstrating how to effectively use UNIX. Some of these activities gave us an introduction to UNIX and its file system. We reviewed articles "Campus Nets for the Nineties" by Raymond K. Neff, Ph.D. and "Technology Across the Campus" on the advances of technology and computer science.

IV. Articles Summaries

"Campus Nets for the Nineties"
by Raymond K. Neff, Ph.D.
Educom Review, Special Issue on Networking
March/April 1996

Case Western Reserve University (CWRU) is upgrading its campuswide networking system by moving from baseband to broadband. They also plan to use upgraded prototypes such as ATM (Asynchronous Transfer Mode) therefore, enhancing its network in terms of the usage of future applications. For example, multimedia data including voice, video and audio can be transmitted on its network.

CWRU has a perception of its campus network contents. First of all, there is a universal network for the campus therefore, everyone has access, utilizing it to its maximum potential. Communications services such as video, voice, multimedia data, and etc. will be supported by its network and the network is fast enough so there is never the problem bottlenecking. Another important aspect of its network is its wire-once architecture, this allows the network cabling to not be reinstalled because of different network topologies that may occur. Mostly, this is due to fiber-optic cabling being used with its longevity and the use single mode and multimode. Single mode is capable of using gigabit and terabit transmission rates while, multimode has can be used as in-building cabling. CWRU also has standards for its signaling and protocols for computer transmission rates which is mostly in part due to ATM and SONET (Synchronous Optical NETwork). They are ran on fiber-optic wiring being that has high scalability speed and ultrahigh-speed transmission.

The university plans to keep up with the changing technology by first going from baseband to broadband. Baseband technology, such as Ethernet, handle single communications channel on a single wire. A broadband technology uses a single wire to transmit multiple

channels of information. They also hope that ATM and possibly SONET will be the preferred transmission technology so that large quantities of data can be packetized. Multimedia applications will be transmitted at the appropriate time so that the problem of segmented or jerky will not exist. The library and classrooms of the future being accessed from a computer pose a big question for the campus network. Since, digital books and images, software libraries and journals are being added to libraries and videoconferencing being one example will help bring the classrooms to the student instead of vice versa show the importance of the campus network and how it will play a big role in the institution's future. By the end of this century, Case Western Reserve University plans to have a new utility infrastructure for communications technology and it also plans to extend beyond the university into the community.

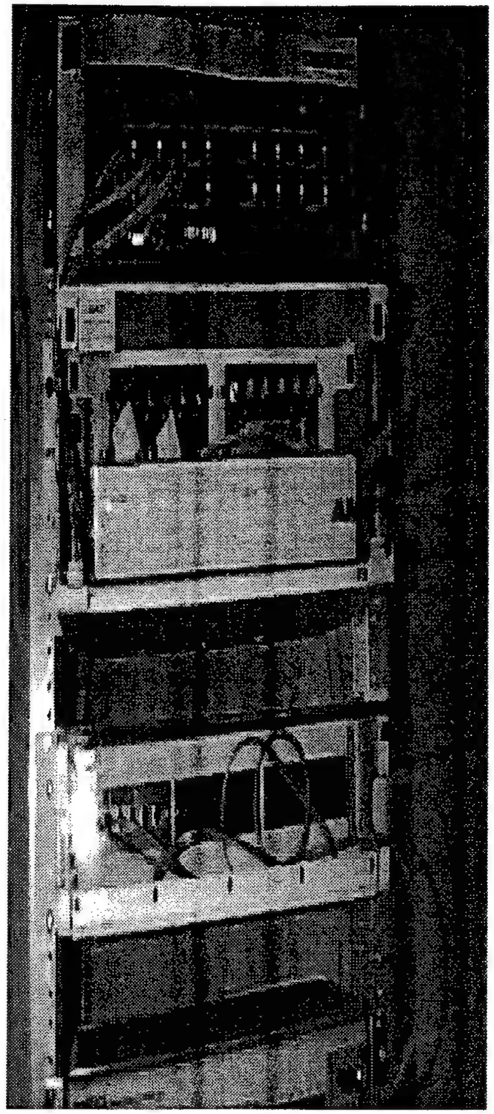
"Technology Across the Campus"
Syllabus 1996

"Technology Across the Campus" discusses technology activities such as a virtual theater, video conferencing, distance learning via optic fiber, and full motion video occurring at four universities across the nation. The University of Kansas uses the virtual theater headed by Mark Reaney, Associate Professor of Theater and Film. He uses virtual reality software Virtus WalkThrough Pro to plan sets for plays. A video device is used to display the background and other images on a screen behind the actors which is monitored and controlled by an offstage computer operator. One aspect that adds to the plays is the use of 3D glasses that see converged dual images giving the illusion of 3D space.

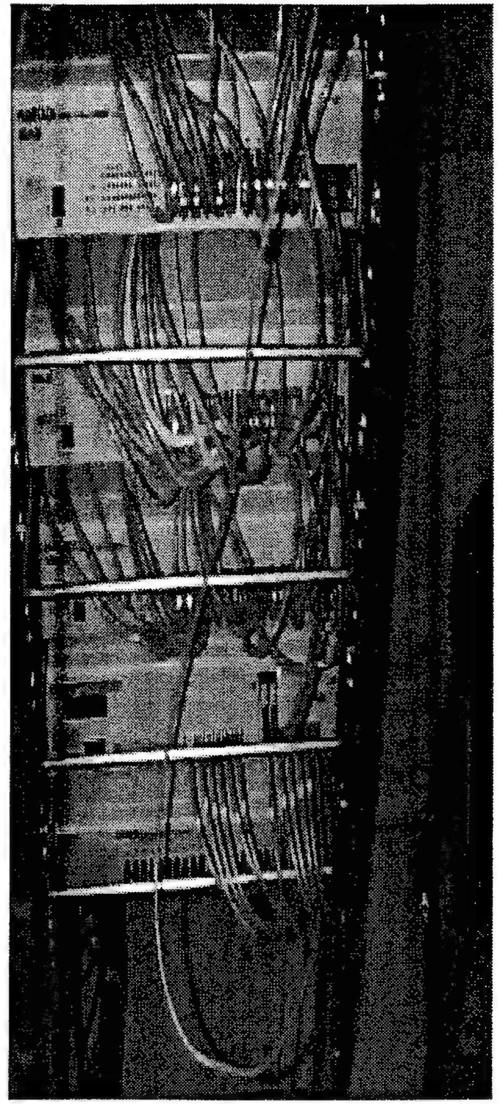
At WSU (Washington State University) video conferencing is used provided to people across the state. In 1985, a program called Washington Higher Education Telecommunications Systems (WHETS) to allow students to take classes held at other locations. This is serviced by VideoServer's Multimedia Conference Servers (MCS) due to its multipoint capabilities. Its network is connect through a microwave LAN-based network. WHETS is proving to be effective because ten years ago only ten students were enrolled now 77 classes with 2,300 students are apart of the program. WSU allows the video conferencing to be utilized for other programs at other institutions such as Spokane Intercollegiate Research and Technical Institute and Seattle Central Community College.

Asbury Theological Seminary uses full-motion in the classroom such as distance learning, video, production studios, and laptop computers to communicate with its students. Each classroom is equipped with a video information and monitor or projection system connected

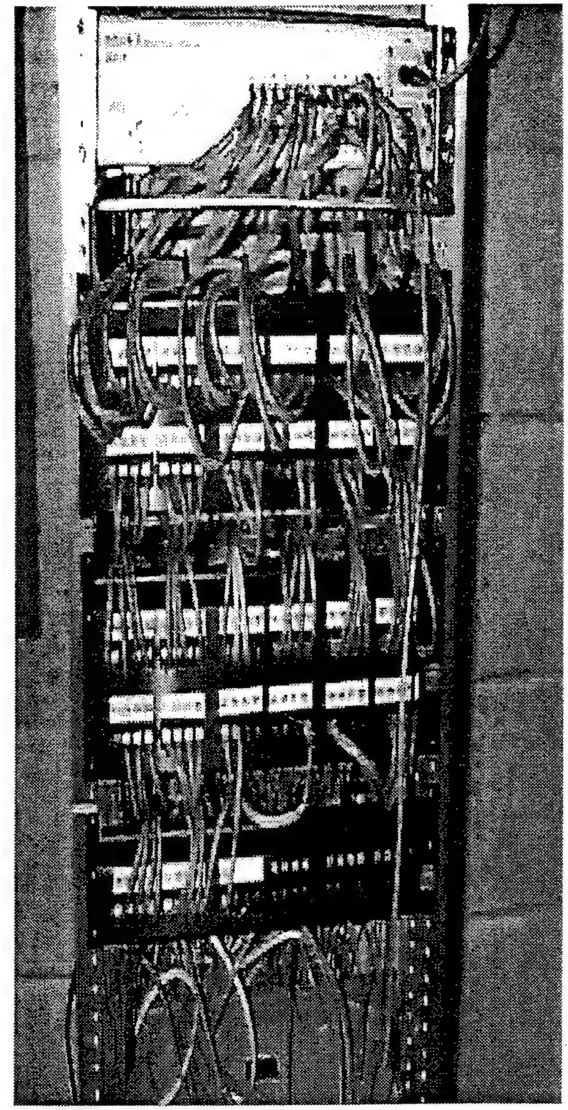
via optic fiber. Asbury operates 48 classrooms spreading over 14 buildings and its distance learning reaches far away as Estonia and India. Southwestern Oklahoma State University also is using distance education over an optical fiber network including its two campuses, two high schools, a junior college and a vocational technical center. The optic fiber network was implemented mostly in stabilize its declining population which has effected its educational system making it hard to fill teaching positions. Therefore, distance education allows resources such as teachers to be shared. These are some of the profiles of technology across the nation allowing other campuses to learn and implement.



ATM Switch and FDC



Rack of Hubs



Punch Down Box

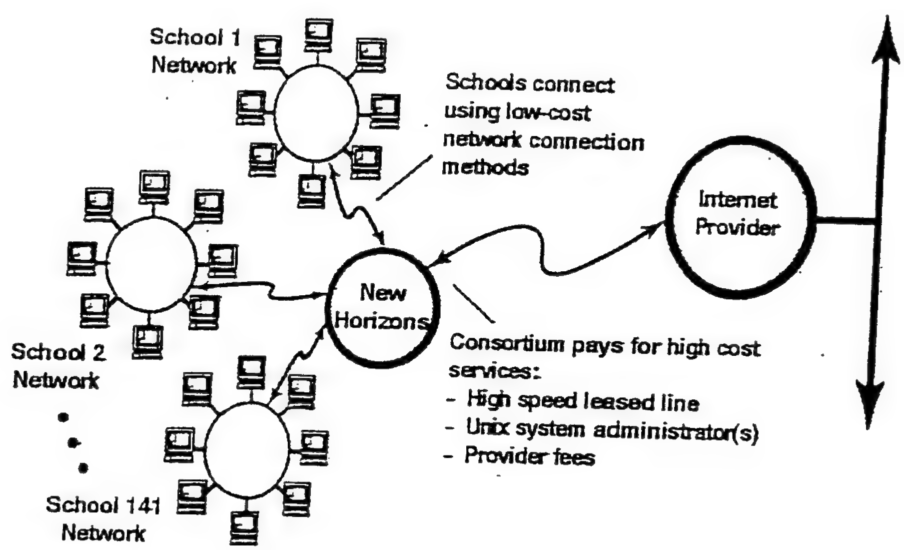
APPENDIX A
(ATM)

APPENDIX B

(ATLAS)

The Wide Area Network

(Using a central site as a connection hub)

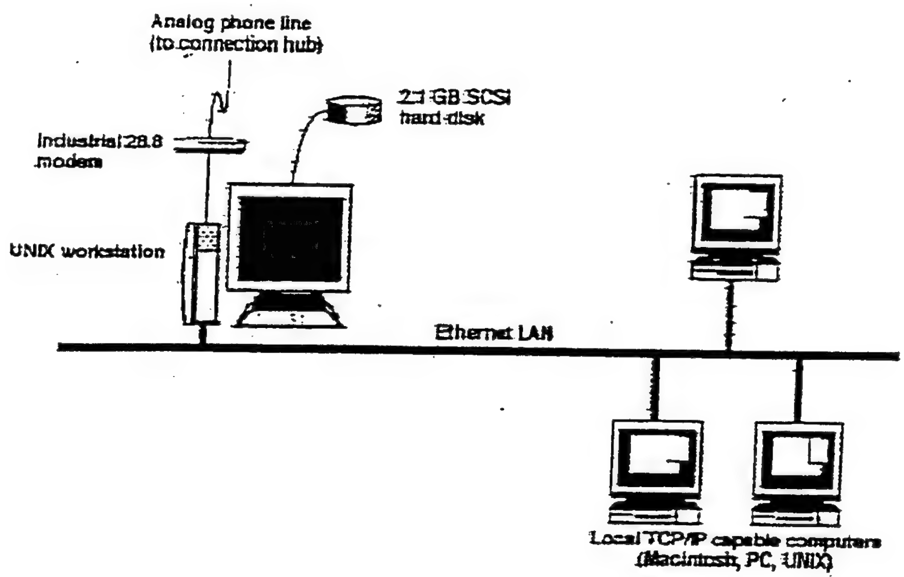


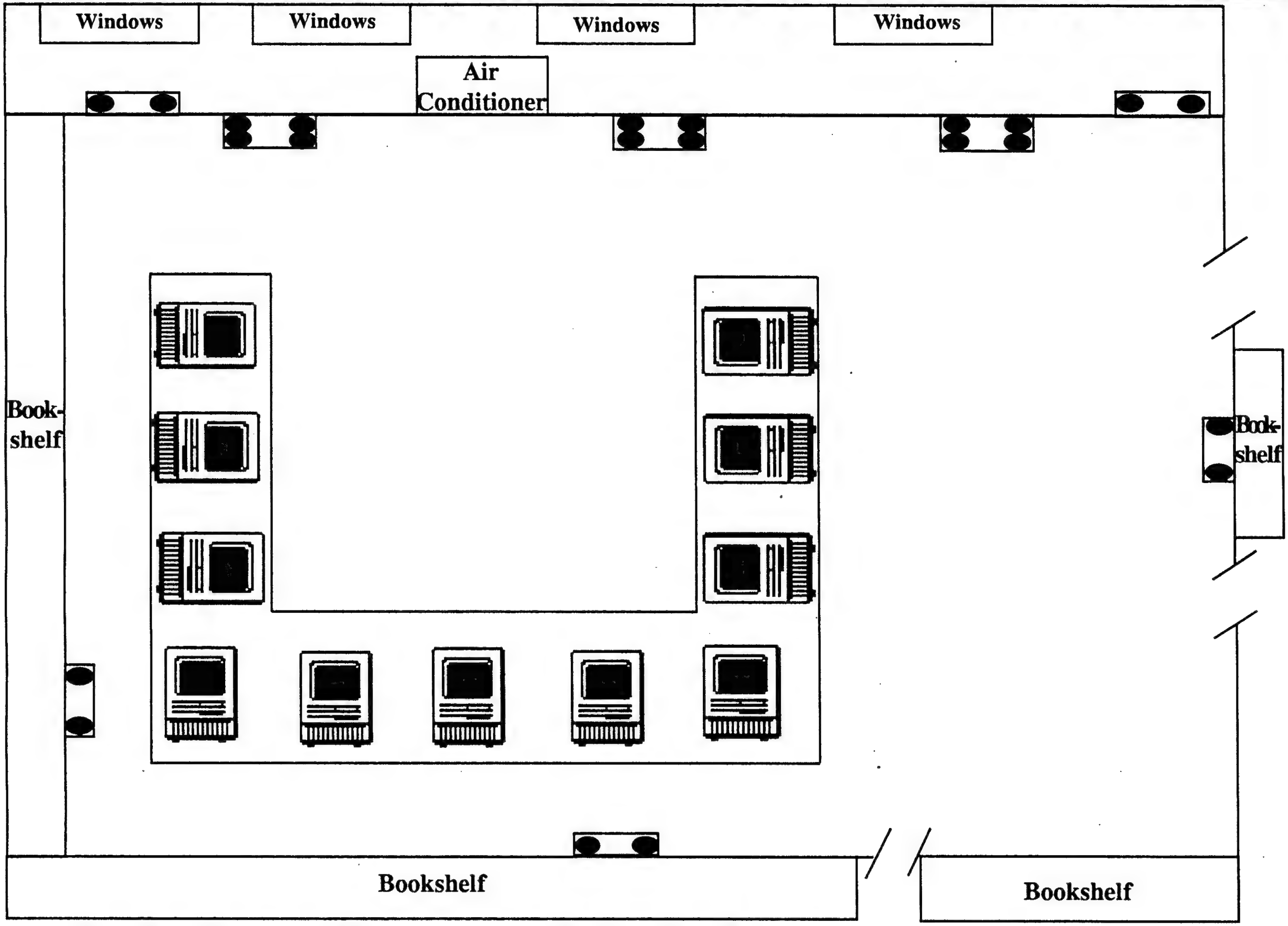
APPENDIX C

(K-12 COMPUTER LAB DIAGRAMS)

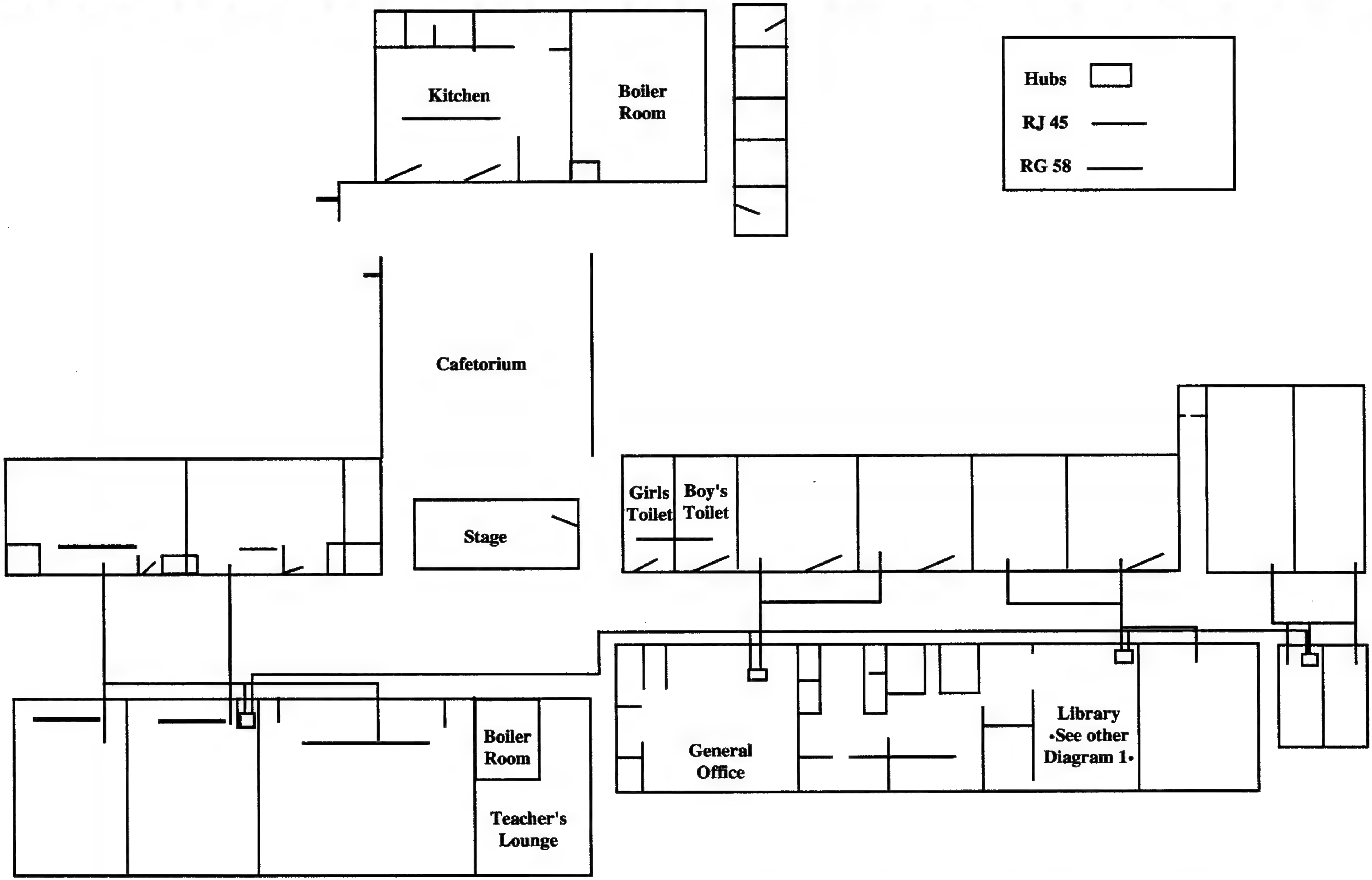
The Local Area Network (LAN)

(The network inside your school building)



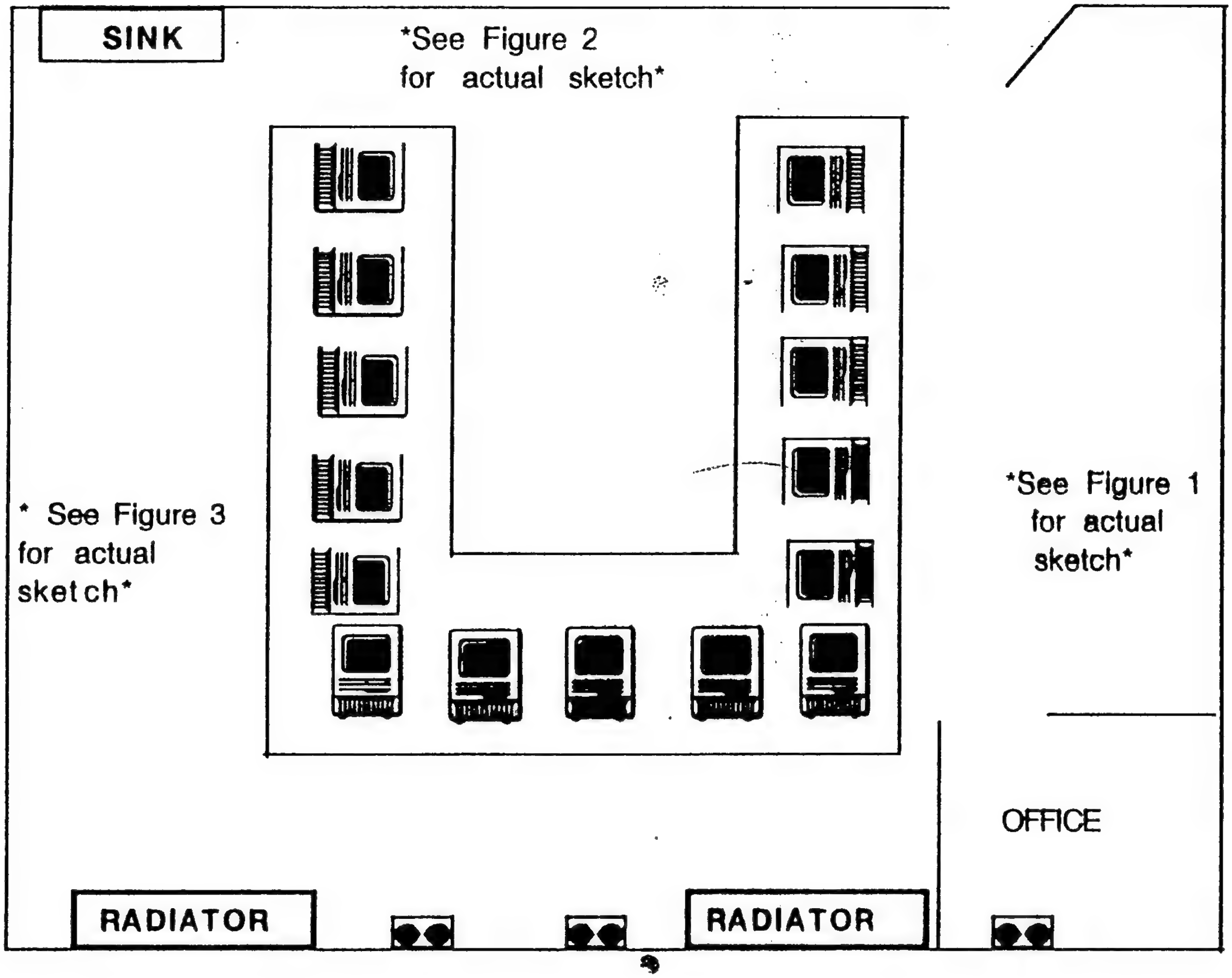


Emily Spong Elementary School's Library



Emily N. Spong Elementary School
Current Floor Plan

DOUGLAS PARK ELEMENTARY



DOUGLAS PARK ELEMENTARY

ACTUAL SKETCH OF WALLS IN ROOM 229

Figure 1

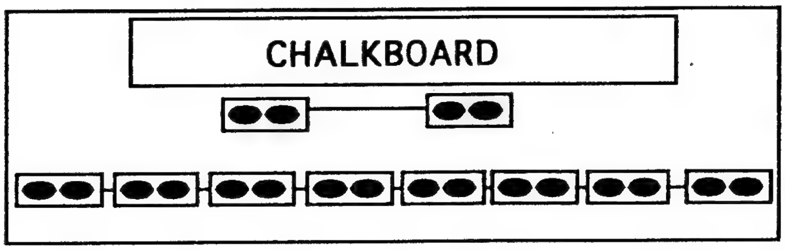


Figure 2

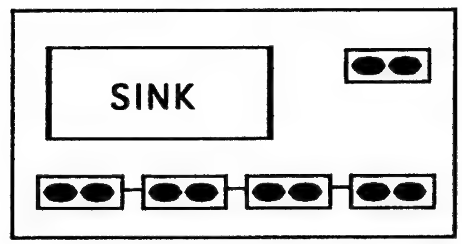


Figure 3

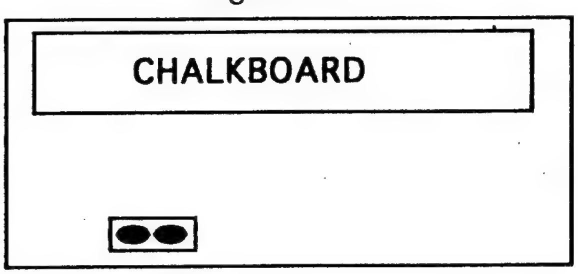
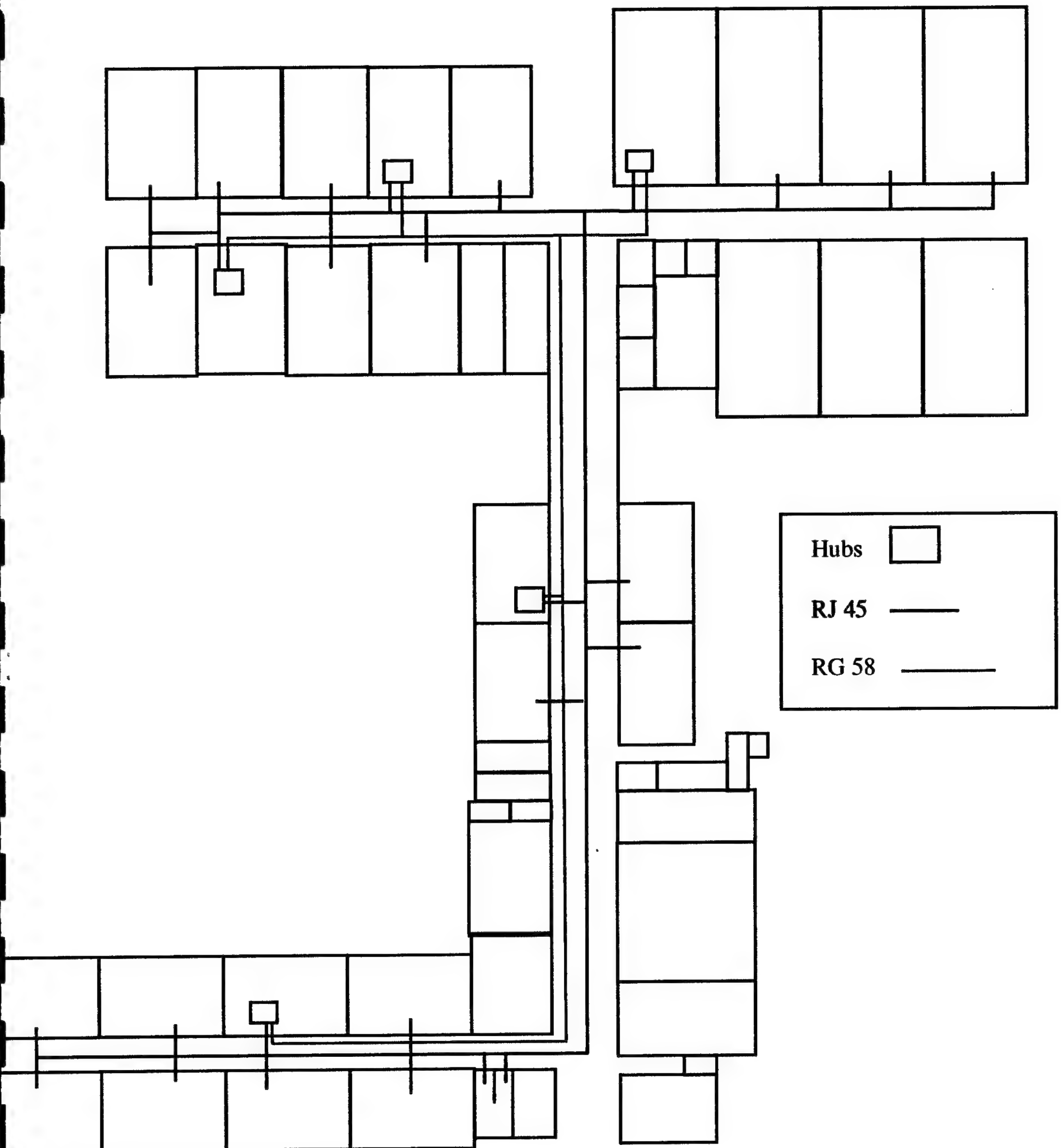


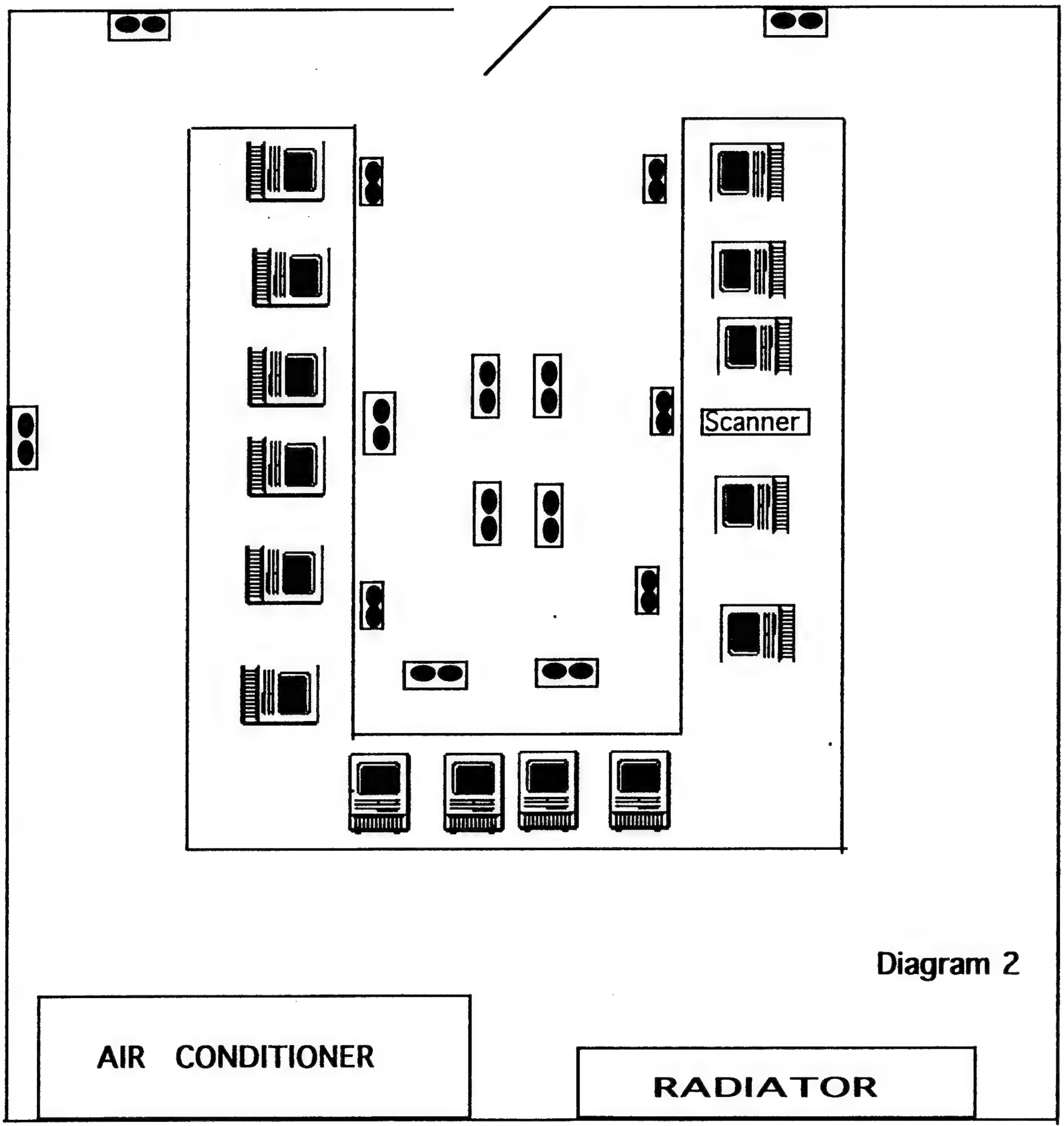
Diagram 2a

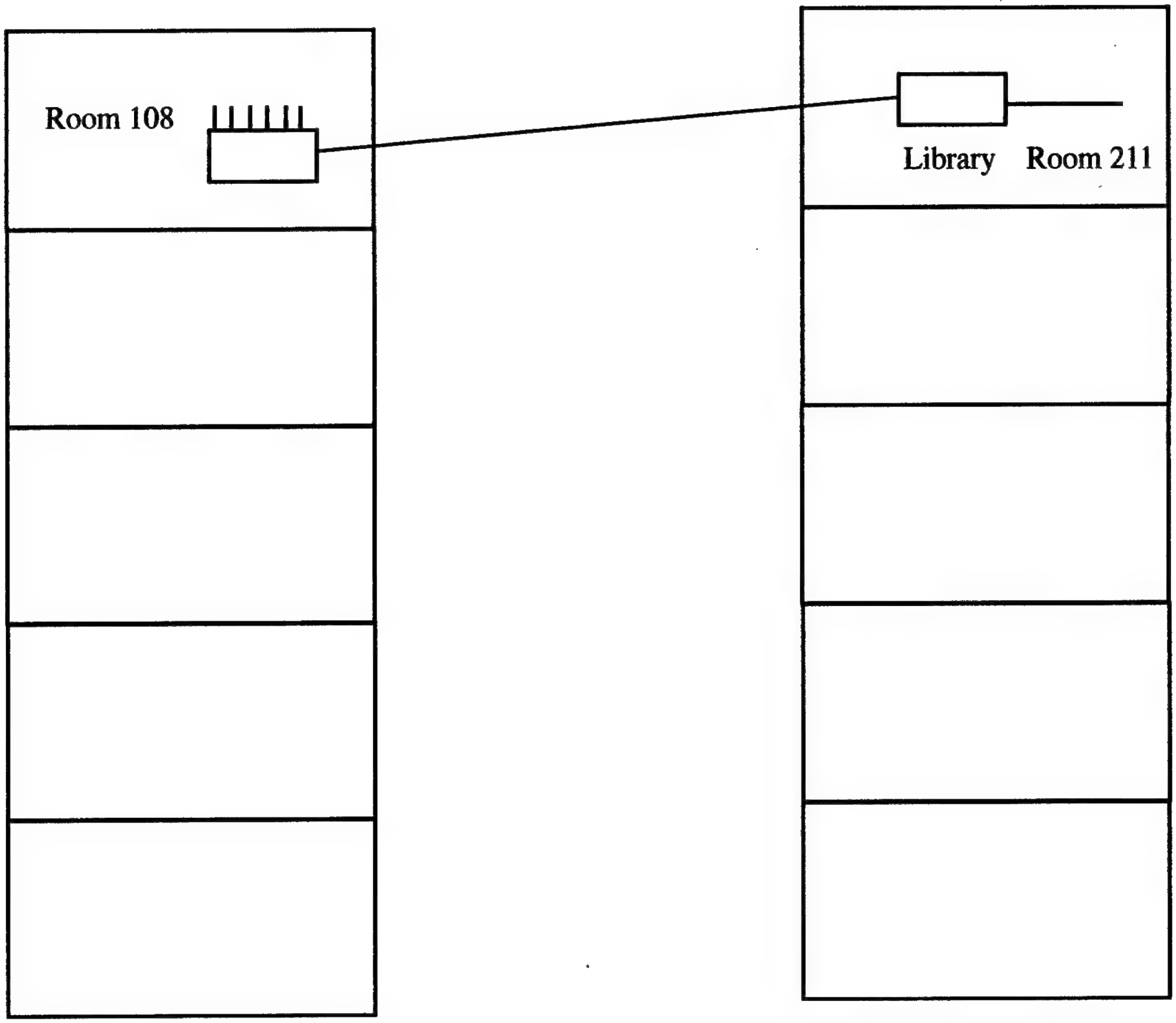


Douglas Park Elementary School

I.C. NORCOM HIGH SCHOOL

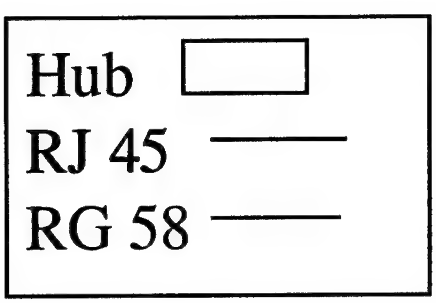
Room 108





Left Wing
Right Side
1st Floor

Left Wing
Left Side
2nd Floor



I. C. Norcom High School Brief Floor Plan

References:

Crisp, Vicki K. and Jeffrey M. Seaton. "Affordable Technology to Link America's Schools". April 1996.

Dyson, Peter. Novell's Dictionary of Networking Novell Press and SYBEX Inc.1994.

Handel Rainer, Manfred N. Huber, and Stefan Schroder. ATM Networks. pp. 23, 61, 89. Addison-Wesley Publishing Co.

FORE SYSTEMS, INC. *ForeRunner* GIA-200 ATM GIO Bus Adapter for Silicon Graphics User's Manual . Section1 p.1. FORE SYSTEMS, INC.

Loomis, Mary E. S. Data Communications. Prentice-Hall, Inc. 1983.

Schaevitz, Alan. "ATM Basic Mechanics", Introduction to ATM.

Stallings, William Local Networks: An Introduction. Macmillian Publishing Co., 1984.

Troy, Douglas. UNIX Systems. The Benjamin / Cummings Publishing Company, Inc., 1990.

Warren, Gary P. and Jeffrey M. Seaton. "A Comprehensive and Cost-Effective Computer Infrastructure for K-12 Schools". pp. 1-7, April 1996.

Wilson, Rodney. UNIX *Tamed* .Prentice-Hall, Inc. 1996.

Appendix and Signature Sheets

Signature Page

1996 AASERT SUMMER RESEARCH PROGRAM

Sponsored by the Office of Naval Research and Elizabeth City State University

Signature Page

1996 AASERT SUMMER RESEARCH PROGRAM

Sponsored by the Office of Naval Research and Elizabeth City State University